Chapter 1

Method

1.1 Causal discovery

In this section, we shall discuss the method for network deconvolution, originally proposed by [?]. The underlying problem is inferring direct effects and dependencies. From this, using prior information on the production setup, we shall be able to infer causal dependencies by directing the resulting edges from the network deconvolution (ND) algorithm. Particularly, the framework and general algorithm proposed Feizi et al. stems from a graph-theoretic approach to the problem of inferring direct dependencies. Namely, suppose that observations are made of some properties of in this case a chemical process. We shall represent these properties as vertices (nodes) V and dependencies between properties as edges. Initially, when observing the vertices, we observe both direct and indirect effects. Particularly, a vertex v_1 might influence some other vertex v_3 through another vertex v_2 if v_2 depends on v_1 and v_3 of v_2. In this case, we will observe that v_1 influences v_3, but actually it is v_2 that has a direct influence on v_3. In graph-theoretical terms, we thus observe the transitive closure of the information that flows between vertices but want to infer the underlying network structure.

An important note on the algorithm to come is that we only use vertices that we have observed. Namely, the underlying structure might be as in Figure 1.1a with an unobserved node/variable (named U in this case). However, without any more assumptions or modelling choices we would (ideally) infer the network structure depicted in Figure 1.1b. With these initial comments, we proceed with the general setup and assumptions for network deconvolution based on observations.

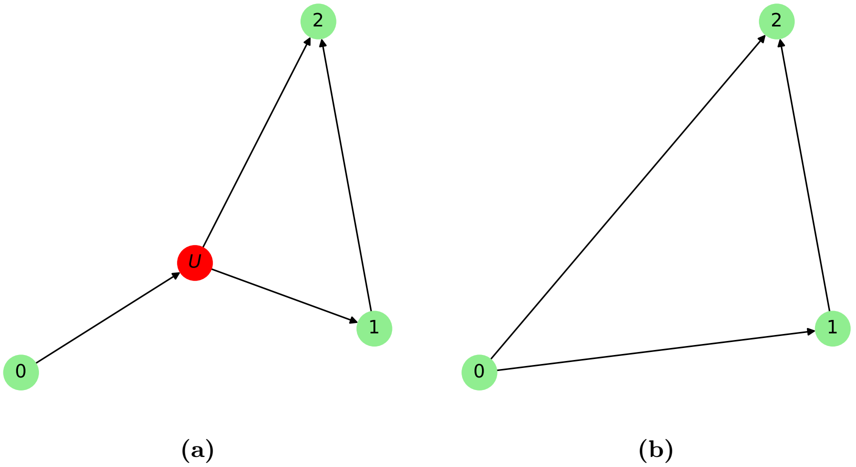


Figure 1.1: (a) An example of a causal structure depicted as a graph. When observing the network, only nodes 0, 1 and 2 are observed/recorded. (b) The resulting inferred graph from observational data. Although this is not a complete picture of the true underlying dynamics of the system, if only the observed variables are of interest, this will be an equally proper representation of the system. Furthermore, in practice this means no further assumptions are made which can and can not be of desire. Namely, if prior information is accessible one might introduce new nodes in the inferred network.

1.1.1 Setup and assumptions

Suppose a set of N random variables (X_i) is given. The method presented in this section aims to discover direct relationships between pairs X_i and X_j for $i \neq j$. These relationships will be presented by an undirected graph as in the previous section. In particular, we shall let each random variable X_i be represented by a vertex in a graph. We will later discuss a way of directing edges such that a causal network may be discovered i.e. a directed acyclic graph that may be used for inference.

The method proposed by [?] then works as follows. Given an observed matrix $G_{obs} \in \mathbb{R}^{N \times N}$ of similarities between each pair of variables, we shall deduce a matrix $G_{dir} \in \mathbb{R}^{N \times N}$ of direct similarities between each pair of random variables X_i and X_j. The measure of similarity, can in practice be any desired measure such a correlation, mutual information which we will focus on in this thesis. See Section 1.2 for a further discussion on these two measures. Note that the algorithm presented will in theory work for non-symmetric measures as well such as *Interaction information*, *Directed information* and *Normalized information.*

The (direct) network is then presented by the discovered G_{dir} containing only the direct effects i.e. interaction between pairs of variables which can be viewed as weights on the edges of the complete graph with nodes representing the random variables. As we shall see in section Subsection 1.3.3, the algorithm is somewhat robust to noise in the sense that we can ensure accuracy depending on the level of noise observed present in G_{obs} and on the norm chosen (from a certain, although rather general, set of norms). This hints to that a threshold on the inferred weights on the edges of the network might be a good idea which is further solidified by the facts that often only the most influential variables are of importance when trying to control the process.

The first assumption is that the observed matrix of co-dependence G_{obs} may be expressed as

$$G_{obs} = G_{dir} + G_{indir} \tag{1.1}$$

Namely, that the direct and indirect effects can be added together to get the total and thus observed interdependence between each pair of variables. Often, this is not the case as we shall see later on. However, the error made from this assumption and the ones to be presented seem to be small enough that the discovered network accurately resemble the true underlying network.

The second and final assumption is that the indirect effects G_{indir} can be computed in terms of G_{dir}. Namely, that

$$G_{indir} = G_{dir}^2 + G_{dir}^3 + \dots \tag{1.2}$$

i.e. that the observed *information* exchanged on an edge e_{ij} between nodes X_i and X_j is the sum second, third etc. order effects, each given by the information on the n-path (where n is the order of the (diminishing) indirect effect) again assumed to a sum of products. In other terms, the second order indirect effect between X_i and X_j (given as the (i, j) element of G_{dir}^2) is the sum of products on edges e_{ik} and e_{kj} for all k

$$\left[G_{dir}^2\right]_{ij} = \sum_{k=1}^{N} e_{ik}\, e_{kj}$$

where e_{ij} is the (i, j) element of G_{dir}. Immediately, we observe that e_{ii} is of interest in terms of its physical meaning. The co-dependence between a random variable and itself might be somewhat ambiguous or even undefined depending on the measure. Thus, the meaning of (non-existing) edges e_{ii} will be of interest later on when using the method on controlled cases to see if any sense can be made of these.

Thus, from the above assumptions, it follows that we can express G_{obs} as

$$G_{obs} = G_{dir} + G_{dir}^2 + G_{dir}^3 + \dots \tag{1.3}$$

Hence, for G_{dir} to exist, it must have spectral radius at most 1 as otherwise, the above sum diverges and thus G_{obs} will not exist. I.e. $\rho\left(G_{dir}\right) < 1$. Thus, assuming convergence we can rewrite the infinite series as

$$G_{obs} = G_{dir}\left(I - G_{dir}\right)^{-1} \tag{1.4}$$

It immediately follows that G_{dir} is given by (can be proved by directly inserting the above expression for G_{obs})

$$G_{dir} = G_{obs}\left(I + G_{obs}\right)^{-1} \tag{1.5}$$

Furthermore, if the measure of dependence between pairs of variables is symmetric, then so is G_{obs} and hence diagonalizable by some orthogonal matrix U such that $G_{obs} = U\Lambda_{obs}U^T$ (with the columns of U being eigenvectors of G_{obs}). It follows that G_{dir} can be expressed in a simple (and later computationally efficient) way

$$G_{dir} = U\Lambda_{dir}U^T$$

where $\Lambda_{dir} = \Lambda_{obs}\left(I + \Lambda_{obs}\right)^{-1}$.

We note that from the above one needs $\left(I + G_{obs}\right)^{-1}$ to be well-defined which is equivalent to $-1 \notin \sigma_{G_{obs}}$ i.e. -1 is not an eigenvalue of G_{obs}. To see that this is indeed the case whenever $\rho\left(G_{dir}\right) < 1$, and that $I + G_{obs}$ is thus invertible we use Equation 1.4 and simplify

$$\begin{aligned} I + G_{obs} &= I + G_{dir}\left(I - G_{dir}\right)^{-1} \\ &= \left(I - G_{dir}\right)\left(I - G_{dir}\right)^{-1} + G_{dir}\left(I - G_{dir}\right)^{-1} \\ &= \left(I - G_{dir}\right)^{-1} \end{aligned}$$

which is clearly invertible. Furthermore, we note that under the assumption $\rho\left(G_{dir}\right) < 1$ we can not place any bound on the spectral radius of G_{obs}. Namely, if v is a unit eigenvector of G_{dir} with eigenvalue λ such that $|\lambda| < 1$, then v is also an eigenvector of G_{obs} as

$$G_{obs}v = \sum_{k=1}^{\infty} G_{dir}^k v = \sum_{k=1}^{\infty} \lambda v = \frac{\lambda}{1 - \lambda} v$$

i.e. $\left(\frac{\lambda}{1-\lambda}, v\right)$ is an eigenpair of G_{obs} and since $\frac{\lambda}{1-\lambda} \in (-1/2, \infty)$ for $\lambda \in (-1, 1)$ we can in general not bound the spectral radius of G_{obs}, although we should never observe an eigenvalue equal to or below $-1/2$ (which again proves that -1 is not an eigenvalue of G_{obs}).

Now, before discussing the implementation and analyzing the algorithm both analytically and through examples, we will take a closer look at the similarity measures that are to be used with this method and that in the end will make up the matrix G_{obs}. Namely, *mutual information* and *correlation.*

1.2 Information measures and computation

In this section we discuss two measures that can be used to construct the matrices of codependency from the previous section. Namely, we shall touch on correlation and discuss what one might choose to call Copula-based entropy. However, before discussing Copula entropy (CE) we first need to define what a copula is.

1.2.1 Copula

Given a set of N random variables $X_1, \ldots, X_N$, a copula is loosely speaking a distribution function with support $[0,1]^N$ incorporating the dependence structure between the random variables. Given a joint distribution function F and (invertible) marginals $F_1, \ldots, F_N$ we define a copula C as

$$F(x_1, \ldots, x_N) = \mathbb{P}\left(X_1 \leq x_1, \ldots, X_N \leq x_N\right) = C\left(F_1\left(x_1\right), \ldots, F_N\left(x_N\right)\right)$$

Letting $u_i = F_i\left(x_i\right) \in [0,1]$ it is clear that C is a distribution function as described above [**?**]. Furthermore, it follows that the marginals of C are uniform. We thus define a copula in probabilistic terms as

Definition 1.1 (Copula). *A function $C : [0,1]^d \to [0,1]$ is called a copula if it has uniform marginals and is a distribution function for a d-dimensional random vector $\mathbf{X}$.*

An important and fundamental theorem of copulas for especially continuous random variables where the marginals are also continuous functions is stated by Sklar:

Theorem 1.2 (Sklar's theorem)**.** *For a random vector $\boldsymbol{X}$ with CDF F and univariate marginal CDFs $F_1, \dots, F_d$. There exists a copula C such that*

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \tag{1.6}$$

If X is continuous, C is unique; otherwise C is uniquely determined on the Cartesian product of the ranges of distribution functions F_i, $\prod Ran\,(F_i)$.

Note that the last statement for non-continuous random variables can be made unique by instead using subcopulas, a generalization of copulas with domain I only a subdomain of the unit hypercube $\mathbb{I}^d = [0,1]^d$ containing all faces of the unit hyper cube. However, there are infinitely many ways of extending such a subcopula to a copula C[**?**]. In our case, this means that for discrete and/or mixed variables, we will later have to work around this non-uniqueness when calculating mutual information. The example made by Geenens[**?**] is a bivariate random vector of independent variables $X \sim \text{Bern}\,(\pi_X)$ and $Y \sim \text{Bern}\,(\pi_Y)$. The support of F_X and F_Y is then $\{0, 1-\pi_X\}$ and $\{0, 1-\pi_Y\}$ respectively. Due to the restriction on the boundary of the unit square, the only unique point of a copula C is then $(1-\pi_X, 1-\pi_Y)$, and by independence we must have

$$C\,(1-\pi_X, 1-\pi_Y) = (1-\pi_X)(1-\pi_Y)$$

Geenens then proceed to define an uncountable set of copulas that fulfill the above criterion which further illustrates that the basic concepts of copulas are not well suited for discrete random vectors. Note that in the article it is however argued how one can extend the concept to a more general concept that works for mixed variables.

From Equation 1.6 we see that a copula is thus simply just a function that *couples* the marginals of a random vector to the joint distribution. The following corollary follows immediately

Corollary 1.2.1 (Coordinate transformation)**.** *Under the assumptions of Theorem 1.2, given any set $(T_1, \dots, T_d)$ of strictly increasing functions, if C is a copula of $(X_1, \dots, X_d)$ then it is also a copula of $(T_1(X_1), \dots, T_d(X_d))$.*

Proof. Suppose $(X_1, \dots, X_d)$ permits a copula C and let T_i be given as stated. Consider coordinate wise the result of the transformation $Y_i = T_i(X_i)$ and consider the CDF $F_{Y_i}(y_i)$

$$F_{Y_i}(y_i) = \mathbb{P}\,(Y_i \leq y_i) = \mathbb{P}\left(T_i^{-1}(Y_i) \leq T_i^{-1}(y_i)\right) = \mathbb{P}\,(X_i \leq x_i) = F_{X_i}(x_i)$$

Thus

$$\begin{aligned} F_{\boldsymbol{X}}(x_1, \dots, x_d) &= C\,(F_{X_1}(x_1), \dots, F_{X_d}(x_d)) \\ &= C\,(F_{Y_1}(y_1), \dots, F_{Y_d}(y_d)) \\ &= F_{\boldsymbol{Y}}(y_1, \dots, y_d) \end{aligned}$$

where Sklar's theorem have been used for the final equality. □

The above corollary is actually equivalent with a seemingly stronger statement and follows easily

Proposition 1.3. *Since T_i is strictly increasing, the inverse T_i^{-1} exists and is also strictly increasing. Thus, the above implication is bidirectional and hence for strictly increasing functions T_i, C is a copula of $(X_1, \ldots, X_d)$ if and only if it is a copula of $(T_1(X_1), \ldots, T_d(X_d))$.*

casuality svarer til at lave nedre/øvre trekant. Er der forskel i at gør edet før og efter for en symmetrisk matrix?

1.2.2 Mutual information and Copula entropy

In this section we introduce Copula entropy as done in [?] and see how it actually is equal to the well known mutual information. The name comes from the general definition of (differential) entropy as we shall see shortly. However, first we define mutual information between a set of random variables

Definition 1.4 (Mutual information)**.** *For a discrete random vector $\boldsymbol{X} = \{X_i\}$, we define the mutual information as*

$$I(\boldsymbol{X}) = \sum_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \log_b \left(\frac{f(\boldsymbol{x})}{\prod_i f_i(x_i)} \right)$$

where $\mathcal{X}$ is the domain of the random vector $\boldsymbol{X}$ and f is the joint probability mass function with marginals f_i. Similarly, for continuous random vectors with f the joint probability density function we define

$$I(\boldsymbol{X}) = \int_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \log_b \left(\frac{f(\boldsymbol{x})}{\prod_i f_i(x_i)} \right) d\boldsymbol{x}$$

where the base of the logarithm b is often chosen to be 2, e or 10 although the choice is unimportant as all logarithms are equivalent up to a scaling factor.

As we shall see later, the continuous version is the limit of the discrete version

We immediately proceed with defining both entropy and differential entropy

Definition 1.5 (Entropy)**.** *The (joint) entropy of a discrete random vector $\boldsymbol{X}$ is defined as*

$$H\left(\boldsymbol{X}\right) = -\sum_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})\,\log_b f(\boldsymbol{x})$$

Definition 1.6 (Differential entropy)**.** *The (joint) differential entropy defined for a continuous random vector $\boldsymbol{X}$ is defined as*

$$h(\boldsymbol{X}) = -\int_{\boldsymbol{x}\in\mathcal{X}} f\left(\boldsymbol{x}\right)\,\log_b f\left(\boldsymbol{x}\right)\,d\boldsymbol{x}$$

We shall see shortly that these are not the limit of each other (discrete to continuous) but when using MI it does not matter and even makes implemtaiton simpler

$$\begin{aligned} I\left(\boldsymbol{X}\right) &= \int_{\boldsymbol{x}\in\mathcal{X}} f\left(\boldsymbol{x}\right)\,\log_b f\left(\boldsymbol{x}\right)\,d\boldsymbol{x} - \sum_{i=1}^{d}\int_{\boldsymbol{x}\in\mathcal{X}} f\left(\boldsymbol{x}\right)\,\log_b f_i\left(x_i\right)\,d\boldsymbol{x} \\ &= \int_{\boldsymbol{x}\in\mathcal{X}} f\left(\boldsymbol{x}\right)\,\log_b f\left(\boldsymbol{x}\right)\,d\boldsymbol{x} - \sum_{i=1}^{d}\int_{x_i\in\mathcal{X}_i} f_i\left(\boldsymbol{x}\right)\,\log_b f_i\left(x_i\right)\,dx_i \\ &= \sum_{i=1}^{d} h\left(X_i\right) - h\left(\boldsymbol{X}\right) \end{aligned}$$

and similarly for the discrete variant.

where the base b, initially, can be chosen at will as all logarithms are equal up to a scalar multiple of each other.

forskellige baser og egentlig ikke et rigtigt valg. Dog er det forskellig skallering af G obs som vi senere viser har en effekt på højere ordens led. Kommenter på/henvis til dette i senere afsnit.

However, to make the calculations more robust and efficient we will use a closely related measure of dependence, namely CE which is defined as follows

$$CE\left(X_1,\ldots,X_n\right) = -\int\cdots\int_{[0,1]^n} c\left(u_1,\ldots,u_n\right)\log_b c\left(u_1,\ldots,u_n\right)\,du_1\ldots du_n$$

where $c(\cdot)$ is the uniquely defined copula density of the joint distribution $f_{\mathbf{X}}(\cdot)$. We show that the above is indeed equal to the negative mutual information.

First, we realize that from definition,

$$\begin{aligned} c(u_1,\ldots,u_n) &= \partial_{\mathbf{u}} C(u_1,\ldots,u_n) \\ &= \partial_{\mathbf{u}} F\left(F_1^{-1}(u_1),\ldots,F_n^{-1}(u_n)\right) \\ &= f(x_1,\ldots,x_n)\frac{1}{f_1(x_1)\ldots f_n(x_n)} \end{aligned}$$

Thus

$$\begin{aligned} -CE &= \int\cdots\int_{[0,1]^n} c(u_1,\ldots,u_n)\log c(u_1\ldots,u_n)\; du_1,\ldots,du_n \\ &= \int\cdots\int_{[0,1]^n} c(u_1,\ldots,u_n)\log c(u_1\ldots,u_n)\; dF_1(x_1),\ldots,dF_n(x_n) \\ &= \int\cdots\int_{\mathbb{R}^n} \frac{f(x_1,\ldots,x_n)}{f_1(x_1)\ldots f_n(x_n)}\log\left(\frac{f(x_1,\ldots,x_n)}{f_1(x_1)\ldots f_n(x_n)}\right) f_1(x_1)\ldots f_n(x_n)\, dx_1\ldots dx_n \\ &= \int\cdots\int_{\mathbb{R}^n} f(x_1,\ldots,x_n)\log\left(\frac{f(x_1,\ldots,x_n)}{f_1(x_1)\ldots f_n(x_n)}\right) dx_1\ldots dx_n \\ &= I(\boldsymbol{X}) \end{aligned}$$

1.2.3 Entropy and mutual information in the limit

Originally, entropy H is only defined for discrete variables. A closely related but yet very different measure for continuous variables is differential entropy h defined as

$$h(X) = \int_{\mathbb{R}} f(x)\,\log f(x)\,dx$$

Although one may think of this as the limit of (discrete) entropy, this is not the case. Namely, consider the support of $f(x)$ (here assumed to be the entire real line) binned into intervals i.e. a discretization of the continuous random variable X. To make notation simpler, we shall bin into equal-sized intervals of width Δ. Then, for each interval $[i\Delta,(i+1)\Delta)$ for $i\in\mathbb{Z}$, there exists an x_i such that the probability mass on this interval is represented by this x_i:

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)\,dx$$

Clearly, this discretization is a valid distribution as

$$\sum_{i\in\mathbb{Z}} f(x_i)\Delta = \int_{\mathbb{R}} f(x)\,dx = 1$$

and in the limit, as $\Delta \to 0$ we recover the original distribution $f(x)$. However, if we try to calculate the entropy of this discretization, denoted by H^Δ, we get a diverging limit

$$\begin{aligned} H^\Delta &= \sum_{i\in\mathbb{Z}} f(x_i)\Delta \log f(x_i)\Delta \\ &= \sum_{i\in\mathbb{Z}} f(x_i)\Delta \log f(x_i) + \sum_{i\in\mathbb{Z}} f(x_i)\Delta \log \Delta \\ &= \sum_{i\in\mathbb{Z}} f(x_i)\Delta \log f(x_i) + \log \Delta \end{aligned}$$

Clearly, the first summand in the above expresion converges to the differential entropy as $\Delta \to 0$ whereas $\log \Delta \to -\infty$.

A similar argument for the joint entropy between the discretization of X_1 and X_2, denoted by H_{12}^Δ, results in

$$H_{12}^\Delta = \sum_{i,j\in\mathbb{Z}} f\left(x_1^{(i)}, x_2^{(j)}\right) \Delta_1\Delta_2 \log f\left(x_1^{(i)}, x_2^{(j)}\right) + \log \Delta_1 + \log \Delta_2$$

where $x_1^{(i)} \in [i\Delta_1, (i+1)\Delta_1)$ and $x_2^{(j)} \in [j\Delta_2, (j+1)\Delta_2)$ are defined such that

$$f\left(x_1^{(i)}, x_2^{(j)}\right) \Delta_1\Delta_2 = \int_{j\Delta_2}^{(j+1)\Delta_2} \int_{i\Delta_1}^{(i+1)\Delta_1} f(x_1, x_2)\, dx_1 dx_2, \quad \forall i, j \in \mathbb{Z}$$

Note that clearly $\left(x_1^{(i)}, x_2^{(j)}\right)$ exists for all $i, j \in \mathbb{Z}$. Again, the joint entropy diverges however, when computing the mutual information, we see that the diverging terms cancel. Namely, from reference til information ud fra entropy proposition

$$\begin{aligned} I_{12}^\Delta &= H_1^\Delta + H_2^\Delta - H_{12}^\Delta \\ &= \sum_{i\in\mathbb{Z}} f_1\left(\tilde{x}_1^{(i)}\right) \Delta_1 \log f_1\left(\tilde{x}_1^{(i)}\right) + \log \Delta_1 \\ &\quad + \sum_{j\in\mathbb{Z}} f_2\left(\tilde{x}_2^{(j)}\right) \Delta_2 \log f_2\left(\tilde{x}_2^{(j)}\right) + \log \Delta_2 \\ &\quad - \sum_{i,j\in\mathbb{Z}} f\left(x_1^{(i)}, x_2^{(j)}\right) \Delta_1\Delta_2 \log f\left(x_1^{(i)}, x_2^{(j)}\right) - \log \Delta_1\Delta_2 \\ &\to h(X_1) + h(X_2) - h(X_1, X_2) \text{ as } \Delta_1, \Delta_2 \to 0 \end{aligned}$$

Thus, the limit of the mutual information for discrete random variables is indeed the mutual information defined for continuous random variables and can be computed either as the limit of discretizing the probability density function and then computing entropies or just using the initial definition for mutual information definition til mutual information.

At this point, we only need G_{obs} and as mentioned earlier we will use mutual information. ref til afsnit om CE Hence, we obtain G_{obs} from the pairwise copula entropy (CE)

$$G_{obs} = \begin{bmatrix} 0 & -CE_{12} & \dots & -CE_{1n} \\ -CE_{21} & 0 & \dots & -CE_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -CE_{n1} & -CE_{n2} & \dots & 0 \end{bmatrix}$$

Although theoretically $-NCE_{ii} = \infty$ for all i, we put 0 in the diagonal because we do not want self explanation as these are trivial. The argument for calculating CE instead of MI are due to the finite volume integral and simpler integrand. In particular, using the copulas, we avoid the fraction $\frac{f(x_1,\dots,x_n)}{f_1(x_1)\dots f_n(x_n)}$ which could easily result in numerical instability e.g. when both f and f_is are close to 0.

Finally, from the deconvoluted information matrix D_{dir} we may choose a threshold t for choosing which edges are significant. The choice of t

1.2.4 Correlation

introducer correlation hurtigt og kommenter på at det kan bruges som codependence mål og derefter ikke kan regnes ud fra copula

In this section we show that in general, the copula does imply correlation. Namely, given a copula C for some set $\{X_i\}_{i\in I}$ indexed by I, one can not calculate ρ between any pair (X_i, X_j), $i \neq j$. This is easily shown by the following argument.

First, note that from Corollary 1.2.1, C is also a copula for $Z_i := (X_i - \mu_i)/\sigma_i$ for $\in\in I$ where $\mu_i = \mathbb{E}[X_i]$ and $\sigma_i = \sqrt{\operatorname{Var} X_i}$. Clearly, the correlation coefficient for Z_i and Z_j is the same as between X_i and X_j. We thus proceed trying to calculate the correlation between any pair Z_i and Z_j.

$$\begin{aligned} \rho_{ij} &= \int\int_{\mathbb{R}^2} z_i z_j f_{ij}(z_i, z_j)\, dz_i\, dz_j \\ &= \int\int_{[0,1]^2} F_i^{-1}(u_i)\, F_j^{-1}(u_j)\, c_{ij}(u_i, u_j)\, du_i\, du_j \end{aligned}$$

where c_{ij} density version of the copula defined for X_i and X_j and F_i and F_j are the marginals of Z_i and Z_j with mean 0 and variance 1. From the above, it is then clear for a fixed, non-constant copula C, the correlation depends on the marginals of X_i and X_j. Also, we see that a constant copula density (only admissible if $c \equiv 1$ on $[0,1]^2$ and 0 elsewhere) always results in $\rho_{ij} = 0$ as

$$\int_0^1 F^{-1}(u)\, du = \int_{\mathbb{R}} z f(z)\, dz = 0$$

again, under the assumption that Z_i has mean 0.

Thus, we conclude that indeed mutual information and correlation is very different measures of codependency (as correlation depends on the marginals whereas mutual information does not) and that it does not make much sense to introduce copulas in the setting of correlation.

1.3 Copula based network discovery

Algorithm 1 G_{obs} computation

```
Require: $N > 0$                                  ▷ Number of variables
  for $1 \leq i < j \leq N$ do
      Estimate $F_i$ and $F_j$ from $x_i^{\mathcal{D}}$ and $x_j^{\mathcal{D}}$
      $u_i^{\mathcal{D}} \leftarrow F_i(x_i^{\mathcal{D}})$
      $u_j^{\mathcal{D}} \leftarrow F_j(x_j^{\mathcal{D}})$
      Estimate $C_{ij}$ from $u_i^{\mathcal{D}}$ and $u_j^{\mathcal{D}}$
      Compute $NCE_{ij}$
      $G_{ij}, G_{ji} \leftarrow -NCE_{ij}$
  end for
```

Remark. The $1-\alpha$ quantile, denoted by $G_{[1-alpha]}$ (of the upper triangular matrix), can be computed in many ways. As it is only used to filter the G_{obs} matrix, its precise value does not matter. Only the property α part of the observations are below $Q_{1-\alpha}$ and $1-\alpha$ are above (or equal to) $Q_{1-\alpha}$. This property is for example fulfilled by the quantile function `quantile` from `NumPy` (v. 1.26.4). Thus setting $\alpha = 1$ will result in G_{obs} retaining all entries (except for the diagonal entries).

Remark. The $\beta \in (0,1)$ parameter serves as a sort of regularization. The algorithm above also maps the maximum absolute value of the eigenvalues (the spectral radius) to β as is also pointed out in the implementation of Source code from Nature paper at https://compbio.mit.edu/nd/index.html. The proof of this is shown below

Algorithm 2 (ND) Network Deconvolution

```
Require: $G_{obs}$                                        ▷ Input observational matrix
  $[G_{obs}]_{ii} \leftarrow 0, \ \forall 1 \leq i \leq N$                          ▷ zero-diagonal
  $Q_p \leftarrow G_{[1-\alpha]}$
  Set $G_{obs}$ 0, where $G_{obs} < Q_p$
  Compute eigendecomposition $Q, \Lambda$ of $G_{obs}$
  $\lambda^+ \leftarrow \max\left(\lambda^{\max}, 0\right)$
  $\lambda^- \leftarrow -\min\left(\lambda^{\min}, 0\right)$
  $m^+ \leftarrow \frac{1-\beta}{\beta}\lambda^+$
  $m^- \leftarrow \frac{1+\beta}{\beta}\lambda^-$
  $m \leftarrow \max\left(m^+, m^-\right)$
  $\hat{\Lambda} \leftarrow \Lambda\left(mI + \Lambda\right)^{-1}$
  return $Q\hat{\Lambda}Q^T$
```

Proof. To show that eigenvalues i.e. the diagonal elements of Λ resulting from Algorithm 2 all fall in the interval $[-\beta, \beta]$ (i.e. $\sigma\left(Q\Lambda Q^T\right) \subseteq [-\beta, \beta]$) where at least one λ is mapped to either $-\beta$ or β, first notice that clearly the resulting eigenvalues of $G_{dir} = Q\hat{\Lambda}Q^T$ are clearly given by $\frac{\lambda_i}{m+\lambda_i}$ where $(\lambda_i)_{\{1,\ldots,N\}}$ are the (real) eigenvalues of G_{obs} from the definition of $\hat{\Lambda}$. We will show the above by first considering $\lambda \geq 0$ and $\lambda < 0$.

For $\lambda \geq 0$, clearly $m \geq \frac{1-\beta}{\beta}\lambda^+$, thus

$$\frac{\lambda}{m+\lambda} = \frac{1}{1+m/\lambda} \leq \frac{1}{1+\frac{\lambda^+}{\lambda}\frac{1-\beta}{\beta}} \leq \frac{1}{1+\frac{1-\beta}{\beta}} = \beta$$

where the final inequality follows from $\lambda \leq \lambda^+$. Hence $[0, \lambda^+] \to [0, \beta]$.

Furthermore, for $0 > \lambda \geq -\lambda^-$, note that also $m \geq \frac{1+\beta}{\beta}\lambda^-$. Since $\beta \in (0, 1]$, $m + \lambda \geq \frac{1+\beta}{\beta}\lambda^- + \lambda > 0$ and thus $\frac{\lambda}{m+\lambda} < 0$ which implies

$$-\frac{\lambda}{m+\lambda} \leq \frac{-\lambda}{\frac{1+\beta}{\beta}\lambda^- + \lambda} = \frac{1}{\frac{1+\beta}{\beta}\frac{\lambda^-}{-\lambda} - 1} \leq \frac{1}{\frac{1+\beta}{\beta} - 1} = \beta$$

i.e. $[-\lambda^-, 0) \to [-\beta, 0)$. This shows that indeed all the eigenvalues of G_{dir} is numerically less that or equal to β. Finally, assuming $m \neq 0$ or equivalently that $G_{obs} \neq \mathbf{0}$, either $m = \frac{1-\beta}{\beta}\lambda^+$ (and thus $\lambda^+ \neq 0$ is an eigenvalue of G_{obs}) for which the above shows that indeed λ^+ is mapped to β or $m = \frac{1+\beta}{\beta}\lambda^-$ (and hence $\lambda^- \neq 0$ and thus $-\lambda^-$ is an eigenvalue of G_{obs}) for which $-\lambda^-$ is mapped to $-\beta$. This shows that G_dir indeed has an eigenvalue which numerical value is β. □

1.3.1 KDE methods

Proposition 1.7. *Given a bivariate normal distribution $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where*

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma^2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

Then the mutual information $I(X_1, X_2) = -\frac{1}{2}\ln\left(1-\rho^2\right)$.

Proof. This follows by direct computation Using e.g. that $I(X_1, X_2) = h(X_1) + h(X_2) - h(X_1, X_2)$ □

$$G_{obs} = \sum_{k \geq 1} G_{dir}^k \tag{1.7}$$

1.3.2 Ensuring convergence and the effect of β

From Network Deconvolution - A General Method to Distinguish Direct Dependencies over Networks - Supplementary Notes and their implementation found at their webpage at https://compbio.mit.edu/nd/ there seem to be an inconsistency between code and theory. In this section, we shall thus investigate from where the discrepancy arises. Initially, from the formulation in Equation 1.7, for the right-hand side to converge, it must have spectral radius at most 1 and to ensure convergence, less than 1. In the latter case,

Noget introduktion til at vi først lige kører nogle basale definition afsted, som kommer til at blive brugt senere

Definition 1.8. *Induced Norm.*

A matrix norm $|||\cdot|||$ is said to be induced by the vector norm $||\cdot||$ when

$$|||A||| = \sup_{||x||=1} ||Ax||$$

See [?]

Definition 1.9. *Sub-multiplicative Matrix norm*

A matrix norm $|||\cdot|||$ is said to be sub-multiplicative, if for every $A, B \in \mathbb{F}^{n\times n}$ where $\mathbb{F}$ is either the real or complex field:

$$|||AB||| \leq |||A||| \cdot |||B|||$$

1.3.3 Robustness to noise

They show that the procedure is robust noise by considering that the observed information matrix is influenced by some noise $N \in \mathbb{R}^{n \times n}$ and characterize the noise by its Euclidean norm $||N||_2 := \sup_{||x||_2=1} ||Nx||_2$. They show that

$$||G_{dir} - \hat{G}_{dir}||_2 \leq \gamma + \mathcal{O}\left(\delta^2 + \gamma^2 + \delta\gamma\right)$$

where γ is the spectral norm of N and δ is the spectral norm of $\hat{G}_{obs}$. However, this upper bound can actually be computed when $\delta + \gamma < 1$ and diverges when $\delta + \gamma > 1$. Furthermore, the result can be generalized to other norms than the spectral norm. In particular, the Frobenius norm admits a similar upper bound on the difference. Consider any matrix norm $|||\cdot|||$ for which $|||AB||| \leq |||A||| \cdot |||B|||$. It then follows that

$$\begin{aligned}
\left|\left|\left|G_{dir} - \hat{G}_{dir}\right|\right|\right| &= \left|\left|\left|G_{obs}\left(I + G_{obs}\right)^{-1} - \hat{G}_{obs}\left(I + \hat{G}_{obs}\right)^{-1}\right|\right|\right| \\
&= \left|\left|\left|-\sum_{k\geq 1}\left(-G_{obs}\right)^k + \sum_{k\geq 1}\left(-\hat{G}_{obs}\right)^k\right|\right|\right| \\
&\leq \sum_{k\geq 1}\left|\left|\left|G_{obs}^k - \left(G_{obs} + N\right)^k\right|\right|\right| \\
&\leq \sum_{k\geq 1}\sum_{i=1}^{k}\binom{k}{i}|||N|||^i\,|||G_{obs}|||^{k-i} \\
&= \sum_{k\geq 1}\left(\left(|||N||| + |||G_{obs}|||\right)^k - |||G_{obs}|||^k\right) \\
&= \frac{|||N||| + |||G_{obs}|||}{1 - \left(|||N||| + |||G_{obs}|||\right)} - \frac{|||G_{obs}|||}{1 - |||G_{obs}|||}
\end{aligned}$$

Where the final equality assumes that $|||N||| + |||G_{obs}||| < 1$ and by splitting up the sum into 2 geometric series. However, we also show that the series diverge when $|||N||| + |||G_{obs}||| > 1$ which is not directly apparent as it is a difference geometric series. Namely, by the ratio test, letting $\gamma = |||N|||$ and $\delta = |||G_{obs}|||$

as by [?] supp. notes NetworkDeconvolution-AGeneralMethodtoD...

$$\begin{aligned}\lim_{n\to\infty}\left|\frac{(\gamma+\delta)^{n+1}-\delta^{n+1}}{(\gamma+\delta)^{n}-\delta^{n}}\right| &= \lim_{n\to\infty}\left|\frac{(\gamma+\delta)\left(1+\frac{\gamma}{\delta}\right)^{n}-\delta}{\left(1+\frac{\gamma}{\delta}\right)^{n}-1}\right| \\ &= \lim_{n\to\infty}\left|\delta+\gamma\frac{\left(1+\frac{\gamma}{\delta}\right)^{n}}{\left(1+\frac{\gamma}{\delta}\right)^{n}-1}\right| \\ &= \lim_{n\to\infty}\left|\delta+\gamma\frac{1}{1-\left(1+\frac{\gamma}{\delta}\right)^{-n}}\right| \\ &= |\gamma+\delta| = \gamma+\delta\end{aligned}$$

as $\gamma, \delta > 0$ unless they are the zero-matrix in which case the above is nonsensical from a perspective of interest. The above shows that indeed the above result diverges when $\gamma + \delta > 1$.

Thus, for the spectral norm, denoted by $|||\cdot|||_2$, one simply needs the fact that it is sub-multiplicative (Definition 1.9) which follows from the fact that the spectral norm is induced by the l_2 norm (see Definition 1.8) on $\mathbb{R}$, and hence

$$\sup_{||x||_2=1}||ABx|| \leq ||A||_2 \cdot \sup_{||x||_2=1}||Bx|| = ||A||_2\,||B||_2$$

which by definition means that $||AB||_2 \leq ||A||_2\,||B||_2$. The Frobenius norm $|||\cdot|||_F$ is also sub-multiplicative and depending on the use case may be very useful.

1.3.4 Normal example error

Eksempel medd normal fordeling og fejlen der bliver lavet vha. algoritmen.

Chapter 2

Results

2.1 Examples

In this section, we will investigate how the algorithms Algorithm 1 and Algorithm 2 works in junction and, if so, observe how the algorithm can fail and what may be done to correct such cases. Initially, a few simple examples involving exponentiated multivariate Gaussians $\boldsymbol{Y}$.

Example 2.1. *Exponentiated multivariate Gaussian*

Let us consider a simple case with $\mathbf{Y} = e^{\mathbf{X}}$ *(element wise exponentiation) where* $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$ *where*

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0.9\sigma_1\sigma_2 & 0 \\ 0.9\sigma_1\sigma_2 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}$$

It is clear that to Algorithm 1, the mean is non-important as simply corresponds to a scaling of the Y_i *variables. Furthermore, because of Corollary 1.2.1, theoretically, due to the uniqueness of the Copula* C *(as* $\boldsymbol{Y}$ *is continuous) we should expect near equal or very similar results for* $\boldsymbol{Y}$ *and* $\boldsymbol{X}$ *from Algorithm 1. Additionally, different* σ *corresponds to different scaling of* $\boldsymbol{X}$*, and thus we should* *observe equal or near equal G_{dir} for all $\boldsymbol{Y}$. Initially, we shall see how this hypothesis holds up to the following three examples*

$$\boldsymbol{\sigma} = (0.07, 0.3, 0.9), \quad \boldsymbol{\sigma} = (1, 1, 1), \quad \boldsymbol{\sigma} = (1, 2, 3)$$

In order for the sample size to not influence the results, we simulate a generous number of samples, namely, for the following results we have used $n = 10{,}000$ samples. For $\boldsymbol{\sigma} = (1, 1, 1)$, Algorithm 1 and Algorithm 2 returns the following (using $\alpha = 1$ and $\beta = 0.99$)

$$G_{dir} = \begin{bmatrix} -0.33396 & 0.6660 & 0.02512 \\ 0.6660 & -0.3341 & 0.02730 \\ 0.02512 & 0.02730 & -0.0020583 \end{bmatrix} \tag{2.1}$$

Similarly, for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$:

$$G_{dir} = \begin{bmatrix} -0.3335 & 0.6665 & 0.01414 \\ 0.6665 & -0.3335 & 0.01418 \\ 0.01414 & 0.01418 & -0.00060124 \end{bmatrix} \tag{2.2}$$

Finally, for $\boldsymbol{\sigma} = (1, 2, 3)$:

$$G_{dir} = \begin{bmatrix} -0.1490 & 0.09535 & 0.3599 \\ 0.09535 & -0.2989 & 0.5831 \\ 0.3599 & 0.5831 & -0.4037 \end{bmatrix}$$

For $\boldsymbol{\sigma} = (1, 1, 1)$ and $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ we observe the most resemblance to the Σ, although the resulting G_{dir} deviate in the final column. The difference is likely produced by Algorithm 1 as if the resulting G_{obs} was the same, then so would G_{dir} and from the above argument, we know that theoretically this should be the case. For the final example, $\boldsymbol{\sigma} = (1, 2, 3)$, we see a completely different result and immediately suspect that there must be some numerical errors. Investigating the partial results of Algorithm 1 we immediately see a flaw in the supposedly uniform variables U_i as shown in figure Figure 2.1

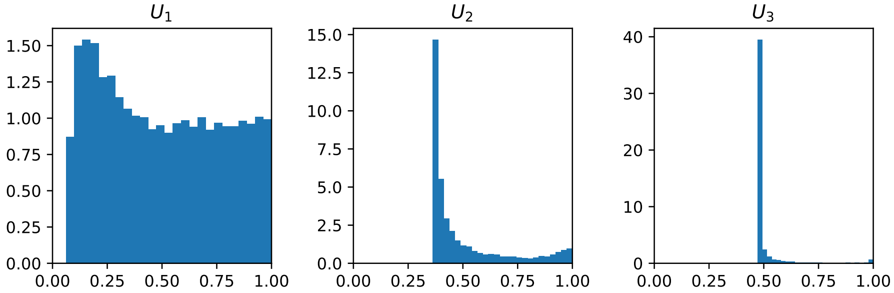


Figure 2.1: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 2, 3)$. These should be uniformly distributed, but clearly this is not the case for U_2 and U_3. Even U_1 does not quite resemble 10,000 samples from a uniform distribution.

Before handling this, the non-uniformity of U_1 in Figure 2.1 is likely also present in the case when $\boldsymbol{\sigma} = (1,1,1)$. Indeed, Figure 2.2 shows that this is indeed the case.

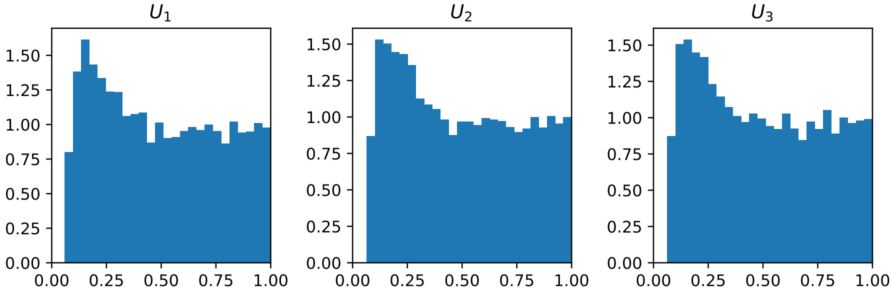


Figure 2.2: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1,1,1)$.

Finally, just to be sure, $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ is also shown in Figure 2.3 and seems very reasonable, except for U_3.

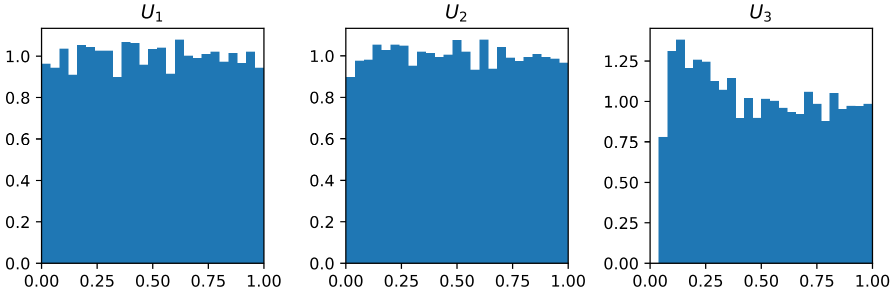


Figure 2.3: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

From the above examples, it seems that the larger the variance, the worse the uniforms turn out. Reasons for this could include numerical issues when trying to calculate $u_i^{(j)}$ form $y_i^{(j)}$ by $u_i^{(j)} = \int_{-\infty}^{y_i^{(j)}} f_i(y)\,dy$ and bad fitting of the kernel density estimate from observations. In particular, for values similar, which happens in the case for large σ such that we observe large negative realizations of X_i, $y_i^{(j)}$ are almost 0, and when computing the integral could result in identical values. Furthermore, from Figure 2.4 we see that indeed the fit is quite poor. Note that we have zoomed in on the interval $[-200, 200]$ which contains 96.2% of *observations. The poor fit is primarily due to the use of Scott's Rule as discussed above which in this case overshoots the optimal bandwidth by a lot.*

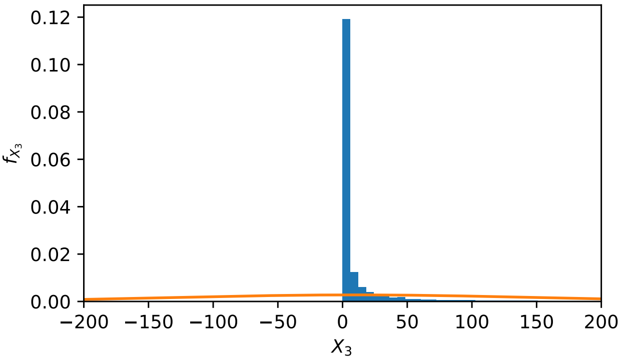


Figure 2.4

The poor fit also explains the high concentration of U_3 around 0.5 in Figure 2.1 as only 54.5% of the probability mass lies above 0.

However, also here Corollary 1.2.1 proves to be useful. Namely, we can get rid of the numerical issues by transforming Y_i using e.g. $\log(\cdot)$ or $(\cdot)^p$ for $p > 0$ to get even out the observations more. As the first simply inverts the initial transformation of X_i, we choose the latter as a more interesting case. In particular, choosing $p < 1$ will result in a more even distribution. In the following, $p = 1/10$ has been used to transform $\boldsymbol{Y}$ prior to running Algorithm 1 and the resulting $u_i^{(j)}$ is shown in Figure 2.5.

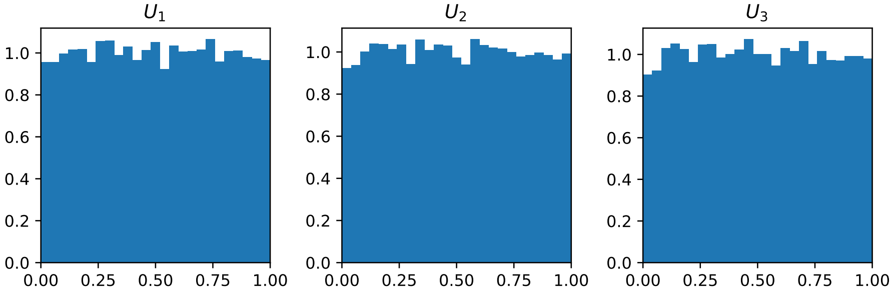


Figure 2.5

The resulting $u_i^{(j)}$ now seem to follow a uniform distribution and indeed the KDE fits much better as seen in Figure 2.6.

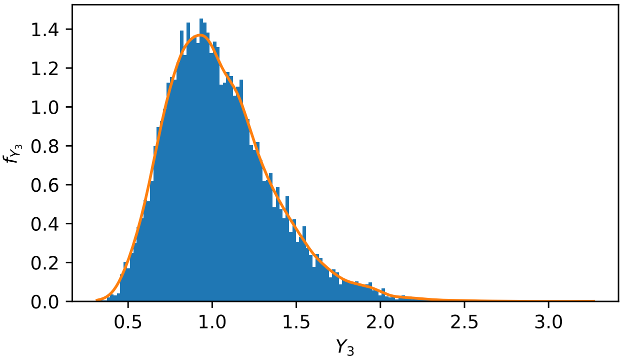
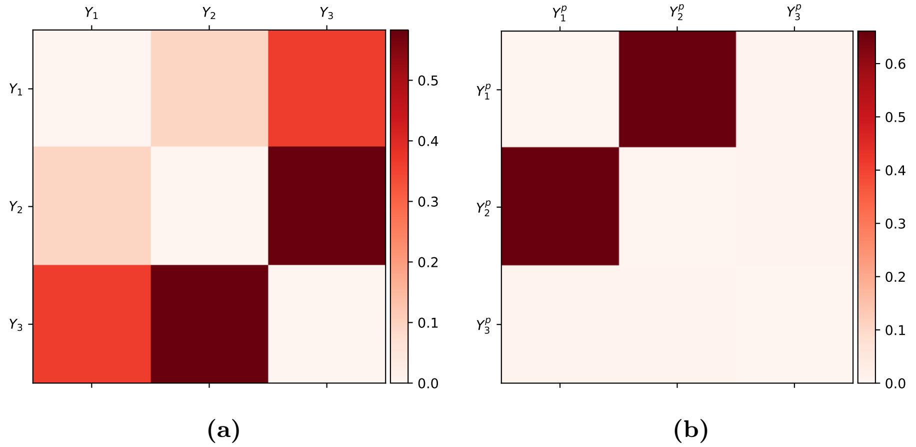


Figure 2.6

Turning to Algorithm 1 and Algorithm 2 we now find that G_{dir} is given by

$$G_{dir} = \begin{bmatrix} -0.3290 & 0.6610 & 0.008440 \\ 0.6610 & -0.3290 & 0.008150 \\ 0.008440 & 0.008150 & -0.0002061 \end{bmatrix}$$

Which is indeed much more comparable with the result from before in Equation 2.1 and Equation 2.2. The difference between G_{dir} from $\boldsymbol{Y}$ and $\boldsymbol{Y}^p$ is clearly visible in Figure 2.7 and also Figure 2.7b resembles the original correlation structure.



(a) **(b)**

Figure 2.7: G_{dir} resulting from 10,000 samples from multi variate Gaussian with $\boldsymbol{\sigma} = (1, 2, 3)$ in **(a)** with raw samples from $\boldsymbol{Y}$ and in **(b)** the transformed data corresponding to $\boldsymbol{Y}^p$.

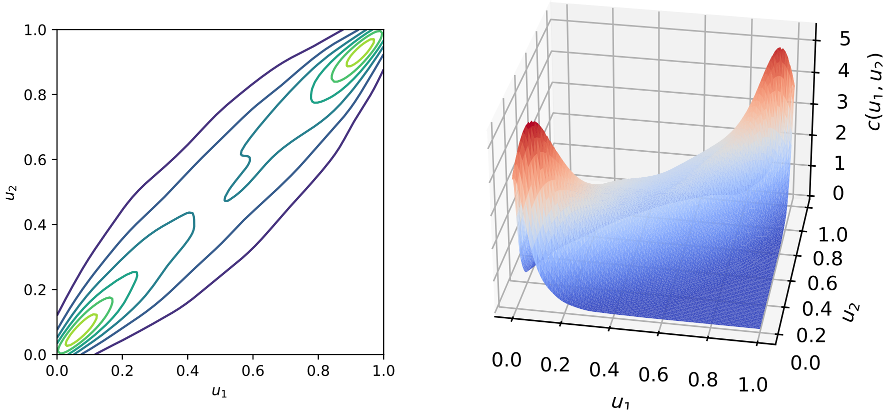
Finally, to end this example we shall compare with some theoretical results. Namely, the output G_{obs} of Algorithm 1 can also be calculated theoretically. For this, we shall use Proposition 1.7 which permits a theoretical result, namely

$$
\begin{aligned}
G_{obs} &= \begin{bmatrix} 0 & -\frac{1}{2}\ln\left(1-\rho_{12}^2\right) & -\frac{1}{2}\ln\left(1-\rho_{13}^2\right) \\ -\frac{1}{2}\ln\left(1-\rho_{21}^2\right) & 0 & -\frac{1}{2}\ln\left(1-\rho_{23}^2\right) \\ -\frac{1}{2}\ln\left(1-\rho_{31}^2\right) & -\frac{1}{2}\ln\left(1-\rho_{32}^2\right) & 0 \end{bmatrix} \\
&\cong \begin{bmatrix} 0 & 0.83037 & 0 \\ 0.83037 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

Similarly, prior to deconvolution, using just the sampled $\boldsymbol{X}$ (i.e. no exponential transform), Algorithm 1 returns

$$
G_{obs} = \begin{bmatrix} 0. & 0.71841756 & 0.01781815 \\ 0.71841756 & 0. & 0.01769672 \\ 0.01781815 & 0.01769672 & 0. \end{bmatrix}
$$

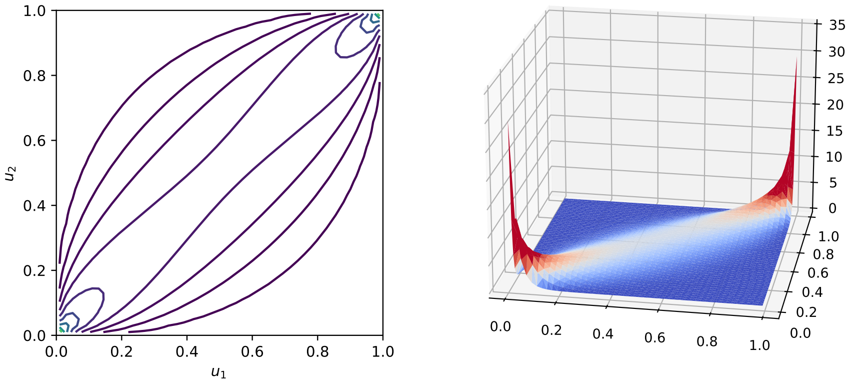
Clearly these are not equal, but in this case, the error is suspected to originate from the estimated joint density. For example, considering X_1 and X_2, we compare the estimated joint copula density and compare to the theoretical reference til et sted hvor gausisk copula står shown in Figure 2.8 and Figure 2.9 respectively.



(a) **(b)**

Figure 2.8: Estimated copula density c with $\rho = 0.9$ corresponding to X_1 and X_2.

The noticeable difference is in the corners $(0,0)$ and $(1,1)$ where the theoretical copula density tends to infinity whereas the estimated density has modes at $(0.1, 0.1)$ and $(0.9, 0.9)$. In particular, simply rescaling the copula density in Algorithm 1 does not resemble the theoretical boundary which is a known issue reference til artikel om undershoot peaks og boundary conditions for KDE. A better approach may be to use jackknifing link til afsnit of jackknifing, som også indeholder reference til artikel hvor dette gøres.



(a) **(b)**

Figure 2.9: Theoretical copula density c with $\rho = 0.9$ corresponding to X_1 and X_2.

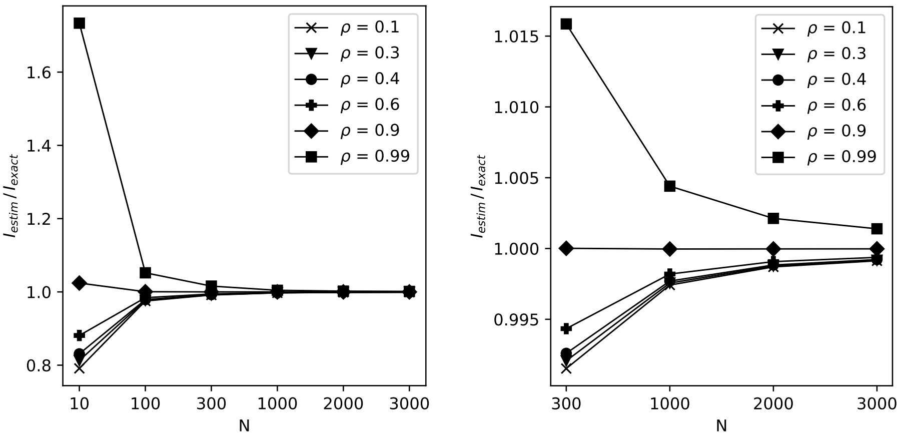
We note however, that the underlying structure is still captured i.e. that Y_1 and Y_2 covary while Y_3 does not inform Y_1 or Y_2 and vice versa.

We continue with a similar example to the previous one. The key difference is the number of variables and a more complicated correlation structure to test the algorithms further.

Example 2.2. *From Example 2.1 we saw how one could handle some numerical issues. Thus, in this example we shall not bother ourselves with such computations and merely focus on the correlation structure. In particular, we shall sample $\boldsymbol{X}$ from a 10 dimensional*

2.2 sammenligning af metoder for at finde MI

Sammenligning af gammel metode og "min"

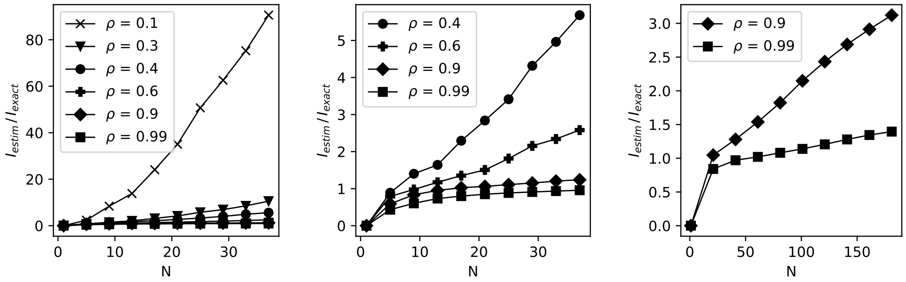


(a)

(b)

Figure 2.10: Evaluation of MI for new method for different N. Bør sammenlignes med artiekl fundet (har sat i bibtex) og original papers (ikke Kina)

Inkluder flere exempler end blot gaussian as done by [?]



(a) **(b)** **(c)**

Figure 2.11: Evaluation of MI for old method for different N. Ligner der er knæk ved fporhold lig 1. Men ved næremere undersøgelse blev det fundet ud af at det ikke helt er tilfældet, og derudover vil der skulle laves en algoritmisk måde at finde dette knæk på. Savitzky–Golay filter kunne være en mulighed, eller gruppere e.g. 5 forskellige bins og tag gennemsnit. Efter smoothing kan anden afledte tæt på 0 bruges, til at finde hvornår stykket bliver fladt (tilnærmelses vist)