CHAPTER 1

Method

1.1 Copula based network discovery

Suppose a set of 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 can be presented using an undirected graph. We will later discuss methods of directing edges such that a causal network may be discovered.

One way of discovering such a network from a set of observations is through mutual information. Namely, suppose that we have obtained a matrix G_{obs} with similarities between each pair of random variables. We will use mutual information as similarity. Let G_{dir} be the information on each edge of the graph i.e. only the direct effects between pairs of variables.

$$G_{obs} = G_{dir} + G_{indir} = G_{dir} + G_{dir}^2 + G_{dir}^3 + \cdots = (I - G_{dir})^{-1} - I = G_{dir} (I - G_{dir})^{-1}$$

where it is assumed that the above infinite sum converges. A necessary and sufficient condition for this is that $\rho(G_d ir) < 1$ where $\rho(\cdot)$ denotes the spectral radius.

It follows that G_{dir} can be calculated as

$$G_{dir} = G_{obs} (I + G_{obs})^{-1}$$

Furthermore, as described by ... . As G_{obs} is symmetric it is diagonalizable with a matrix U such that $\Lambda_{obs} = U^T G_{obs} U$ where Λ_{obs} is a diagonal matrix containing the spectral values with multiplicity in any ordering. It follows that since $\rho\left(G_{obs}\right) < 1$, $\lambda_{obs} + I$ is also invertible and

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

Where $\Lambda_{dir} = \Lambda_{obs}\left(I + \Lambda_{obs}\right)^{-1}$ i.e. $\left[\Lambda_{dir}\right]_{ii} = \frac{\lambda_{obs}}{1+\lambda_{obs}}$ and 0 elsewhere. Whether this is better than a normal inversion of $I + G_{obs}$ is at this point unknown.

At this point, we only need G_{obs} and as mentioned earlier we will use mutual information. 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 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,

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

Thus

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

Hence, we obtain G_{obs} from the pairwise copula entropy (CE)

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

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 $\frac{f(x_1,\ldots,x_n)}{f_1(x_1)\ldots f_n(x_n)}$ 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

Theorem 1.1 (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*

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d))$$

If X is continuous, C is unique.

The following corollary follows immediately

Corollary 1.1.1. *Coordinate transformation*

Under the assumptions of Theorem 1.1, 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

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

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 1.2. *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 $(X_1, \dots, X_d)$ if and only if it is a copula of $(T_1(X_1), \dots, T_d(X_d))$.*

1.2 Algorithms

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
```

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 $Q\hat{\Lambda}Q^T$
```

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

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. $\square$

1.3 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 1.1. *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 1.1.1, 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* $\beta = 0.99$*)*

$$G_{dir} = \begin{bmatrix} -0.33396 & 0.6660 & 0.02512 \\ 0.6660 & -0.3341 & 0.02730 \\ 0.02512 & 0.02730 & -0.0020583 \end{bmatrix} \tag{1.1}$$

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

$$G_{dir} = \begin{bmatrix} -0.3335 & 0.6665 & 0.01414 \\ 0.6665 & -0.3335 & 0.01418 \\ 0.01414 & 0.01418 & -0.00060124 \end{bmatrix} \tag{1.2}$$

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 1.1

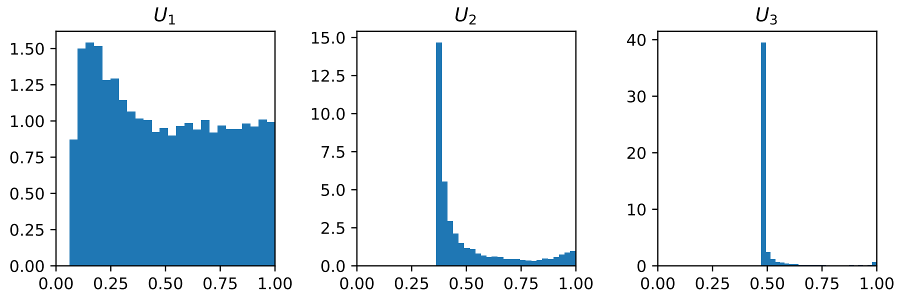


Figure 1.1: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 2, 3)$. These should be uniformly distributed, but clearly this is not the case for U_2 and U_3. Even U_1 does not quite resemble 10,000 samples from a uniform distribution.

Before handling this, the non-uniformity of U_1 in Figure 1.1 is likely also present in the case when $\boldsymbol{\sigma} = (1, 1, 1)$. Indeed, Figure 1.2 shows that this is indeed the case.

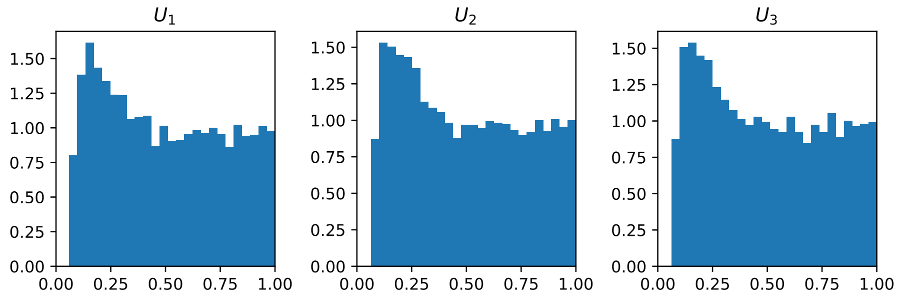


Figure 1.2: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 1, 1)$.

Finally, just to be sure, $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ is also shown in Figure 1.3 and *seems very reasonable, except for U_3.*

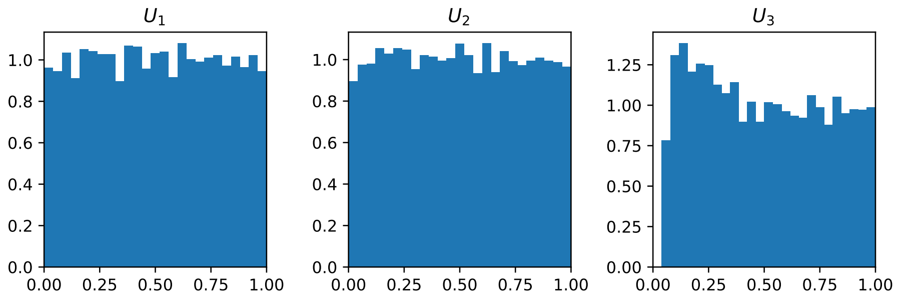


Figure 1.3: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

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 1.4 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.

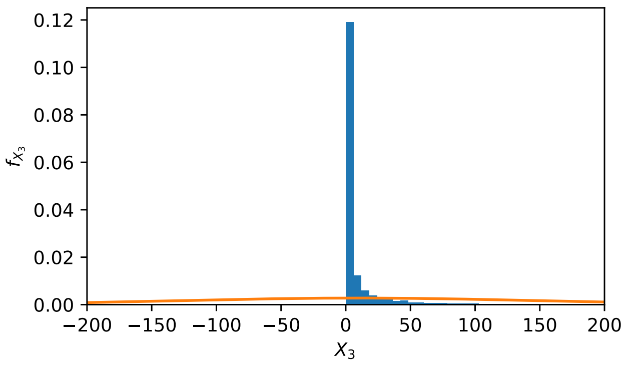


Figure 1.4

The poor fit also explains the high concentration of U_3 *around* 0.5 *in Figure 1.1 as only* 54.5% *of the probability mass lies above* 0.

However, also here Corollary 1.1.1 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 1.5.*

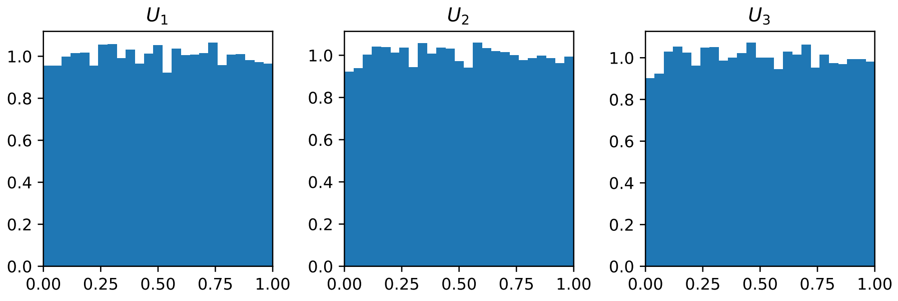


Figure 1.5

The resulting $u_i^{(j)}$ *now seem to follow a uniform distribution and indeed the KDE fits much better as seen in Figure 1.6.*

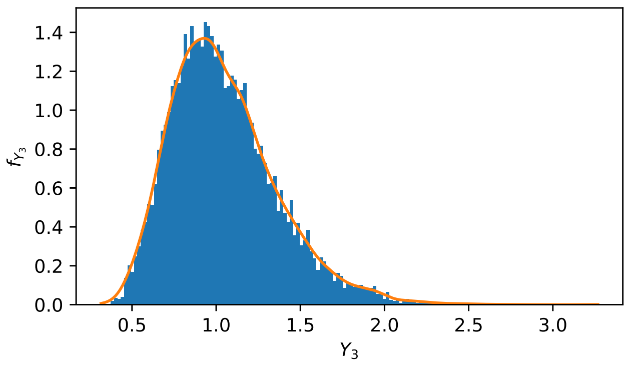
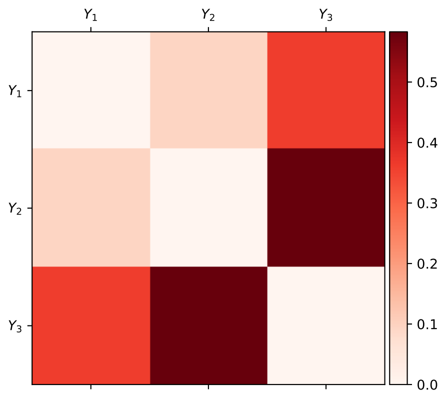


Figure 1.6

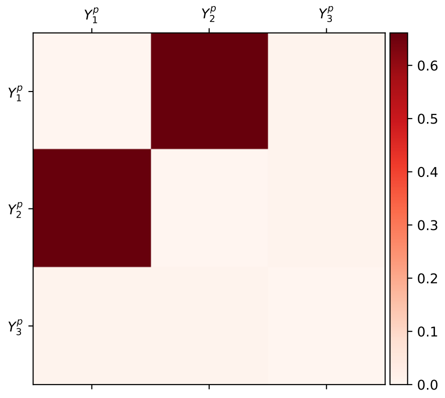
Turning to Algorithm 1 and Algorithm 2 we now find that G_{dir} is given by

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

Which is indeed much more comparable with the result from before in Equation 1.1 and Equation 1.2. The difference between G_{dir} from $\boldsymbol{Y}$ and $\boldsymbol{Y}^p$ is clearly visible in Figure 1.7 and also Figure 1.7b resembles the original correlation structure.



(a)



(b)

Figure 1.7: 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 $\boldsymbol{Y}^p$.

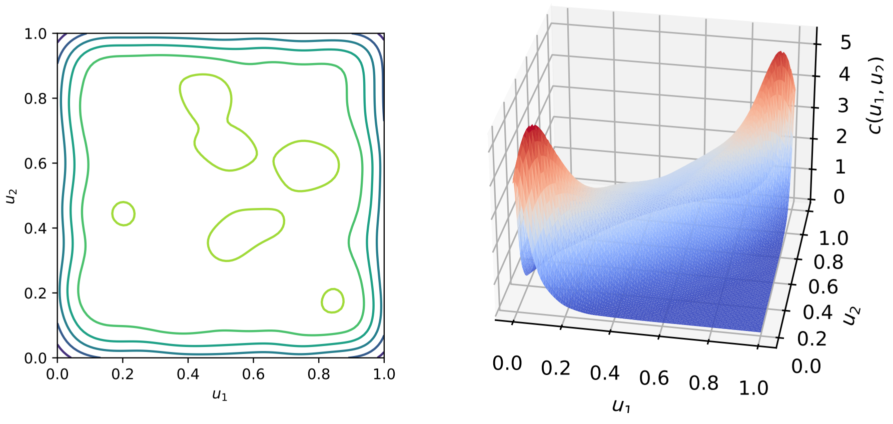
Finally, to end this example we shall compare with some theoretical results. Namely, the output G_{obs} of Algorithm 1 can also be calculated theoretically. For this, we shall use Proposition 1.3 which permits a theoretical result, namely

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

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 1.8 and Figure 1.9 respectively.



(a) **(b)**

Figure 1.8: 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.

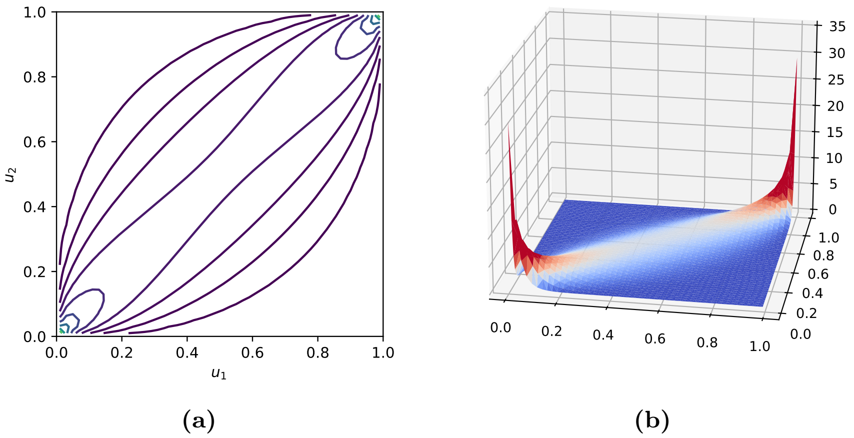


Figure 1.9: Theoretical copula density c with $\rho = 0.9$ corresponding to X_1 and X_2.

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 1.2. *From Example 1.1 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*

Proposition 1.3. *Given a bivariate normal distribution* $\boldsymbol{X} \sim \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right)$ *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 $I\left(X_1, X_2\right) = -\frac{1}{2}\ln\left(1-\rho^2\right)$.

Proof. $\square$