Chapter 1

Results

An introduction to what is going to be included in this section. What results etc.

In this section, we will investigate how the algorithms **??** and **??** works in junction and individually. We shall observe how the algorithms can fail and what may be done to correct such cases.

Overordnet pointe er at genere forskellige mulige graphisce modeller, som senere ville kunne bruges til at lave PGM el.l. Er nok bedst som et ekspolartivt værktøj, og vi undersøger her forskellige situationer, og hvornår der kan ske fejl ud fra om det er lange kæder af kausalitet eller mere komplekse strukturer

1.1 Gaussian chains

In this section we discuss the errors made from the assumption that indirect effects can be computed as a sum of powers of the direct effects, i.e. $G_{indir} = \sum_{k\geq 1} G_{dir}^k$. In particular, on a theoretical level, we shall observe the error in G_{obs} based on the above assumption of how similarities are *convolved* which we equate with the noise N from **??**, although it is a systematic error. To do this, we shall in this section use a multivariate Gaussian to be able to control the correlation and as an extension of this, the mutual information between pairs of random variables. As we already know, correlation and mutual information is independent of the mean and variance of each of the variables however for a bivariate Gaussian the mutual information is given by the correlation as stated in the following proposition.

Proposition 1.1. *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 $I(X_1, X_2) = h(X_1) + h(X_2) - h(X_1, X_2)$ □

Thus, if we know a correlation structure of a Gaussian random vector, we also know the mutual information between every pair of variables which we shall now use in the following made up example. Namely, what we shall denote as a Gaussian chain defined as a Gaussian random vector in the following way. Let $\boldsymbol{X}$ be a d-dimensional Gaussian random vector, the $\boldsymbol{X}$ is a standard Gaussian chain if it can be written in the following way in terms of d independent standard normal variables Z_i up to a permutation i.e. there exists a permutation of the variables of the random vector $\boldsymbol{X}$ that permits the following structure.

$$\begin{aligned} X_1 &= Z_1 \\ X_2 &= \rho_{1,2}X_1 + \sqrt{1-\rho_{1,2}^2}Z_2 \\ X_3 &= \rho_{2,3}X_2 + \sqrt{1-\rho_{2,3}^2}Z_3 \\ &\vdots \\ X_d &= \rho_{d-1,d}X_{d-1} + \sqrt{1-\rho_{d-1,d}^2}Z_d \end{aligned} \tag{1.1}$$

It follows that the marginals have variance 1 as clearly $\mathrm{Var}[X_1] = \mathrm{Var}[Z_1] = 1$ and for $i > 1$, $\mathrm{Var}[X_i] = \rho_{i-1,i}^2 \mathrm{Var}[X_{i-1}] + \left(1 - \rho_{i-1,i}^2\right) \mathrm{Var}[Z_i] = 1$ by independence of X_{i-1} and Z_i. Thus, the above structure also implies the Cholesky factorization of the correlation matrix for $\boldsymbol{X}$, namely

$$L = \begin{bmatrix} 1 & & & & \\ \rho_{1,2} & \sqrt{1-\rho_{1,2}^2} & & & \\ \rho_{2,3}\rho_{1,2} & \rho_{2,3}\sqrt{1-\rho_{1,2}^2} & \sqrt{1-\rho_{2,3}^2} & & \\ \vdots & & & \ddots & \\ \prod_{i=2}^{d} \rho_{i-1,i} & \cdots & \sqrt{1-\rho_{j-1,j}^2} \prod_{i=j+1}^{d} \rho_{i-1,i} & \cdots & \sqrt{1-\rho_{d-1,d}^2} \end{bmatrix}$$

Which will allow us to both sample from such a chain and calculate G_{dir} and G_{obs} theoretically. However, in this example, it is easier to calculate the correlation between the variable X_i and X_j directly. As the variance of each variable is 1 we simply calculate the covariance. We assume without loss of generality that $i < j$ whence

$$\mathrm{Cov}[X_i, X_j] = \mathrm{Cov}\left[X_i, \rho_{j-1,j} X_{j-1} + \sqrt{1-\rho_{j-1,j}^2} Z_j\right] = \rho_{j-1,j} \mathrm{Cov}[X_i, X_{j-1}]$$

which by induction implies $\rho_{i,j} = \prod_{k=i+1}^{j} \rho_{k-1,k} = \rho_{j,i}$. At this point, we are almost ready to use the algorithms from the previous chapter. First, we will only use **??** to deconvolve the network based on theoretical correlations and later mutual information. However, before doing so, we note that from the definition in Equation 1.1 the random variable $\boldsymbol{X}$ exhibits a Markovian property. Namely, the X_i above can be understood discrete stochastic process as they are successively drawn based only on the previous variable X_{i-1} i.e. $f(X_i \mid X_{i-1}, X_{i-1}, \ldots, X_1) = f(X_i \mid X_{i-1})$. Thus, if the algorithm works as intended, we should observe that the deconvolved network is a *chain* of variables as shown in the Figure 1.1 Thus, we now have the expected result, and we

Figure 1.1: The graphical representation of a Gaussian chain. Arrows signify a possible causal structure. If furthermore, one assumes that X_1 is generated first, then X_2 and so on, this is the only causal structure that would make sense.

proceed with using correlation and mutual information to try and rediscover this structure in the following two sections

1.1.1 Gaussian chain deconvolution using correlation

In this section, we will use the observed correlation i.e. $\rho_{i,j} = \prod_{k=i+1}^{j} \rho_{k-1,k}$ for the elements of G_{obs} with $i < j$. Note that although it makes sense to consider the correlation between a variable and itself, we shall as discussed before set the diagonal to 0. Furthermore, we have the choice of using either a symmetrical G_{obs} or a (upper or lower) triangular G_{obs}. We shall first use an upper triangular G_{obs} but before using deconvolving using **??** we note that we can actually get the result theoretically. Also, as G_{obs} is in this case strictly upper triangular, the spectral radius is 0 and hence we have no problems with converge of the infinite sum of powers of (the uniquely defined) G_{dir}. From the above, it is clear that G_{obs} is given as follows

$$G_{obs} = \begin{bmatrix} 0 & \rho_{1,2} & \rho_{1,2}\,\rho_{2,3} & \cdots & \prod_{k=2}^{d} \rho_{k-1,k} \\ & 0 & \rho_{2,3} & \cdots & \prod_{k=3}^{d} \rho_{k-1,k} \\ & & \ddots & & \vdots \\ & & & 0 & \rho_{d-1,d} \\ & & & & 0 \end{bmatrix} \tag{1.2}$$

Now, let G_{dir} be given as follows

$$G_{dir} = \begin{bmatrix} 0 & \rho_{1,2} & & & \\ & 0 & \rho_{2,3} & & \\ & & \ddots & \ddots & \\ & & & 0 & \rho_{d-1,d} \\ & & & & 0 \end{bmatrix}$$

then G_{dir}^2 is given by

$$G_{dir}^2 = \begin{bmatrix} 0 & 0 & \rho_{1,2}\,\rho_{2,3} & & & \\ & 0 & 0 & \rho_{2,3}\,\rho_{3,4} & & \\ & & \ddots & \ddots & \ddots & \\ & & & 0 & 0 & \rho_{d-2,d-1}\,\rho_{d-1,d} \\ & & & & 0 & 0 \end{bmatrix}$$

It is not hard to show that in fact $\sum_{k\geq 1} G_{dir}^k = \sum_{k=1}^{d} G_{dir}^k = G_{obs}$. Thus, if we know a graph topological ordering of the variables corresponding to the structural causal model, we completely recover (without any error) the direct dependencies/correlation from to the initial definition in Equation 1.1. This actually holds for a general *chain* where Z_i can follow any distribution as long as they are independent as the above computations did not use the fact that Z_i follows a standard Gaussian. From this, we might think that if we have a topological ordering of the variables this is the preferred method, and it is as

long as correlation is a good enough measure of similarity/codependency. Albeit this is only shown for the special case of a chain, in Section 1.2 we consider the more general case and conclude that this indeed holds. Regarding the comment on correlation being a good enough measure of similarity, a prototypical case is when joint probability density function of two variables resemble a parabola. Namely, let $X_1 \sim \mathcal{U}(0,1)$ and $X_2 \mid X_1 \sim \mathcal{N}\left(1-4\left(x_1-1/2\right)^2, \sigma^2\right)$ i.e. the joint distribution function is a parabola with a Gaussian noise added along the second dimension. In Figure 1.2, 1000 samples from this distribution is shown for $\sigma = 1/10$ along with the expectation $\mathbb{E}[X_2|X_1]$. It is not hard to show that the covariance between X_1 and X_2 is 0 however we clearly see a relationship between the two variables. In fact, computing the mutual information results in $I(X_1, X_2) \approx 1.030$ implying $X_1 \not\perp X_2$ i.e. there exists a higher order (non-linear) dependency. Thus, if the algorithm permits, we would prefer mutual information to correlation as we can then use observed higher order relationships to infer a causal structure. On a more technical point of view, we note that

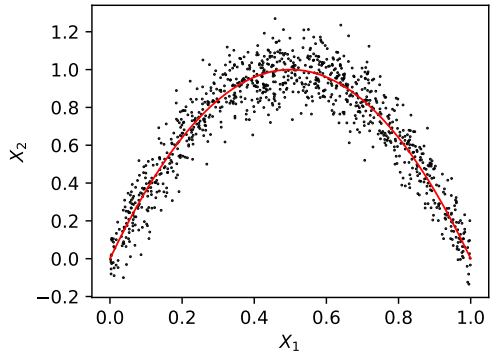


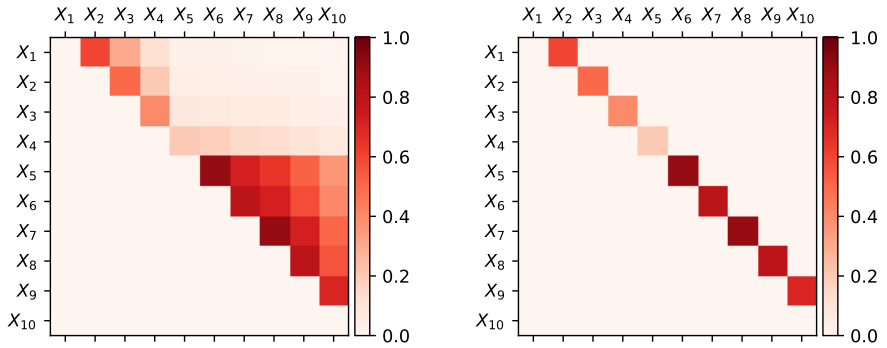
Figure 1.2: 1000 samples generated from $X_1 \sim \mathcal{U}(0,1)$ and $X_2 \mid X_1 \sim \mathcal{N}\left(1-4\left(x_1-1/2\right)^2, \sigma^2\right)$ with $\sigma = 1/10$. The mutual information is calculated theoretically to be $I(X_1, X_2) \approx 1.030$ and repeated simulations show that the empirical correlation is symmetric around 0 supporting the claim that the underlying correlation is in fact 0

mutual information is a measure of how dense the joint distribution is, invariant to scale. In a way, it is a measure of how close the joint distribution is to a lower dimensional manifold.

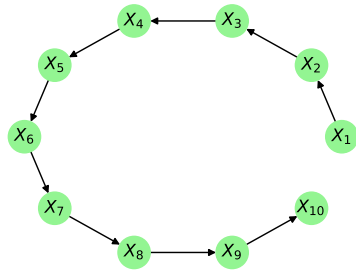
We proceed with a 10-Gaussian chain defined by the following correlations:

$$\begin{aligned} &\rho_{1,2} = 0.6, \quad \rho_{2,3} = 0.5, \quad \rho_{3,4} = 0.4 \\ &\rho_{4,5} = 0.2, \quad \rho_{5,6} = 0.9, \quad \rho_{6,7} = 0.8 \\ &\rho_{7,8} = 0.9, \quad \rho_{8,9} = 0.8, \quad \rho_{9,10} = 0.7 \end{aligned} \tag{1.3}$$

We have chosen correlations of different sizes to check if the deconvolution is robust in presence of both strong and weak links. In particular, X_5 is only $\rho^2_{4,5} = 4\%$ of X_4 and the remaining 96% is noise/indescribable variance i.e. a very weak link between the first part of the chain up to and including X_4 and the rest. However, as discussed above, if let G_{obs} be upper triangular, we should completely rediscover these direct relations which is indeed also the case. In particular, from Figure 1.3 we observe that the inferred network, represented by G_{dir}, is indeed a chain of variables and is exactly equal to the theoretical G_{dir} as we would expect (up to very small rounding errors of the size 10^{-16}). The *estimated* G_{dir} is also shown as a directed graph which the initial topological assumption implies, with edges wherever G_{dir} is non-zero. We now proceed to



(a) G_{obs} **(b)** G_{dir}



(c) G_{dir} as a digraph

Figure 1.3: Results from using an upper triangular G_{obs} and correlation to infer the causal network structure. (a) shows the upper triangular G_{obs} with the correlation between every pair of variables. (b) shows the deconvolved G_{obs} and as we expect, the superdiagonal contains the original correlations given in Equation 1.3. (c) shows G_{dir} represented as a digraph and matches the expected result.

investigate what happens when we remove the prior information of the topological ordering. Namely, if G_{obs} is no longer triangular but symmetric. In particular, let T_{dir} be given as G_{dir} above. We then have that G_{dir} in the symmetric case is $T_{dir} + T_{dir}^T$ and similarly for G_{obs}, $G_{obs} = T_{obs} + T_{obs}^T$. Clearly, $I + G_{obs}$ is positive definite as it is a proper correlation matrix. However, that also implies that we might have eigenvalues of G_{obs} less than or equal to $-1/2$ which we know from **??** is not the result of a G_{dir} such that **??** holds as then the infinite sum diverges. However, as -1 is not an eigenvalue of G_{obs}, we will investigate what happens if one tries to use **??** anyway.

First, we shall however discuss the errors being made using the symmetric G_{obs} and G_{dir}. Namely, we investigate the powers of G_{dir}:

$$G_{dir}^2 = \left(T_{dir} + T_{dir}^T\right)^2 = T_{dir}^2 + \left(T_{dir}^T\right)^2 + T_{dir}T_{dir}^T + T_{dir}^T T_{dir}$$

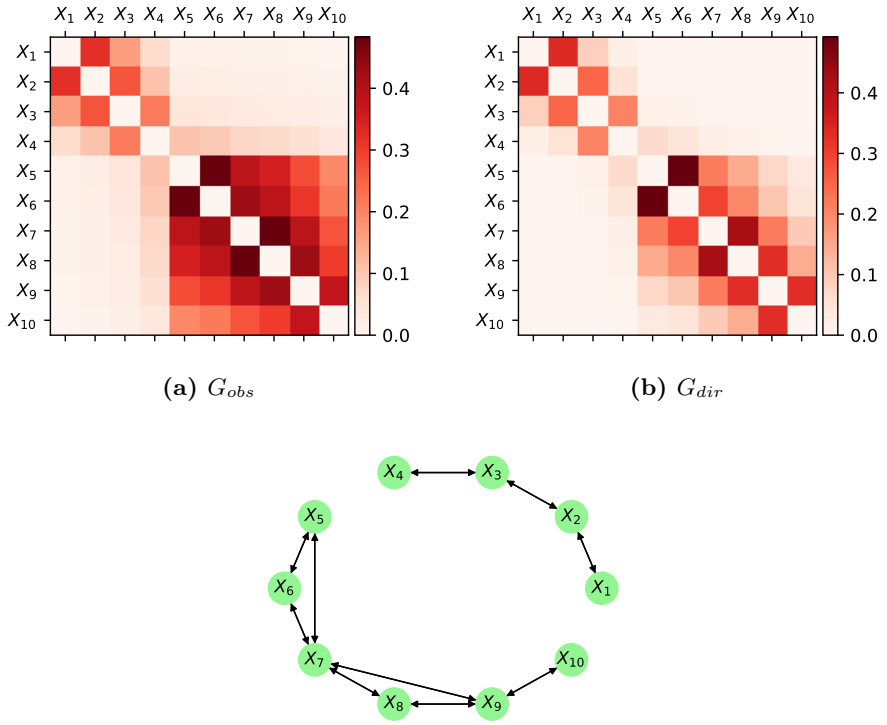
Higher power can be calculated similarly, but for the second power we already observe an error. The first two terms corresponds to a reflection of the second order effects that we saw above and know to be true, whence the final two terms, that add to a diagonal matrix, is an error and will propagate with higher order powers of G_{dir}. Through simple calculation the resulting error is

$$T_{dir}T_{dir}^T + T_{dir}^T T_{dir} = \begin{bmatrix} \rho_{1,2}^2 + \rho_{2,3}^2 & & & \\ & \rho_{2,3}^2 + \rho_{3,4}^2 & & \\ & & \ddots & \\ & & & \rho_{d-2,d-1}^2 + \rho_{d-1,d}^2 \end{bmatrix}$$

Thus, for chains, we expect larger errors for sub-chains with strong links i.e. a subgraph of a chain that is also a chain where the correlation from one variable to the next is large. Using $G_{obs} = T_{obs} + T_{obs}^T$ we have that the smallest eigenvalue is approximately $\lambda_{\min} \approx -0.92263$ thus, multiplying G_{obs} with a constant $c_s < 0.54192$ will make G_{dir} have spectral radius at most 1. The results vary with one or two edges for the choice of c_s and in the following we have chosen $c_s = 0.53651$ resulting in $\rho\left(G_{dir}\right) \approx 0.98020$ and $\tilde{G}_{obs}$ and $\tilde{G}_{dir}$ as seen in Figure 1.4. From Figure 1.4b we see that some correlation/association seem to bleed to variables 2 or 3 edges away which we of course know is not true given the Markov property discussed above. However, it is also clear that the error here is that the original assumption does not hold since using a symmetric G_{obs} imply that the measure of similarity flows both ways where in this case it is very much unidirectional.

From Figure 1.4 conclude that we are somewhat able to rediscover the causal structure. Not surprisingly, we observe that the weak link between X_4 and X_5 is one of the first to break and that we observe some extra edges between the later more strongly linked sub-chain as by the above discussion. Finally, before presenting the results for the unscaled G_{obs} (where the smallest eigenvalue is

smaller than $-1/2$) we note that changing the parameter α in **??** did not have much of an effect indicating that the network is quite sparse (as we also know it to be) as even removing 65% of the smallest correlations from G_{obs} did not have any effect. The chosen threshold of $t = 0.2$ on G_{dir} seemed to be the best compromise of a connected graph and the density of the edges (although this is somewhat biased from prior knowledge of the true graphical structure). Finally, we try using the unscaled G_{obs} in **??**. Interestingly, we find that the



(a) G_{obs}

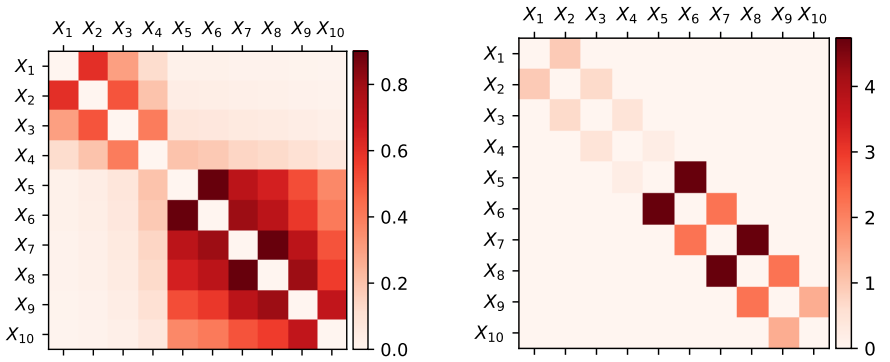
(b) G_{dir}

(c) G_{dir} as a graph

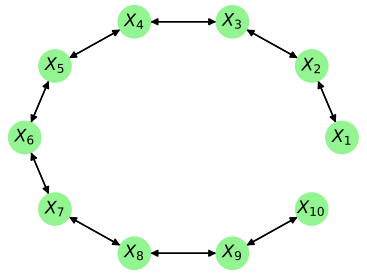
Figure 1.4: Using a symmetric G_{obs} as shown in (a), we observe that higher order effects start to emerge as can be seen in (b). The main response is still in the superdiagonal and subdiagonal as we expect, where some similarity seems to bleed to nearby nodes/variables thus making the threshold used important for the resulting graph. For (c), a threshold $t = 0.2$ was used to obtain a decent compromise between connectedness and denseness of the direct association.

true structure emerges as can be seen from Figure 1.5. Although the *correlations* in Figure 1.5b are not really correlations they do resemble those discovered in Figure 1.3b. On closer inspection, it is not apparent how they are related except

that it is a non-linear relationship. Although in this case it seemed to work not rescaling G_{obs} in order to discover the causal structure we will in general not apply this to real world scenarios as the method is not well-defined in terms of assumptions and what the resulting G_{dir} should be interpreted as. Thus, at



(a) G_{obs} **(b)** G_{dir}



(c) G_{dir} as a graph

Figure 1.5: Using an unscaled (symmetric) G_{obs} results in a good recovery of the causal structure as seen in (b) and (c). However, at this point it is not clear whether it holds only for chains and using correlation or if it is a more general phenomenon.

this point we have a rather good understanding of how the method works on Gaussian chains if one uses correlation as a measure of association. Furthermore, if one knows (a plausible) topological ordering of the variables, we are able to perfectly rediscover the network of direct dependencies. However, as noted above, correlation is not always a good measure of similarity. Thus, we proceed experimenting with mutual information on the same Gaussian chain.

1.1.2 Gaussian chain deconvolution using mutual information

In this section, we continue the example from the previous section but instead of using correlation as a measure of similarity, we will use mutual information. Immediately, we note that mutual information or Copula entropy does not propagate as assumed in **??**. As an example, from Proposition 1.1, we know that the mutual information in the case of a Gaussian chain between a variable X_i and the next variable X_{i+1} is $-1/2 \log\left(1-\rho_{i,i+1}^2\right)$ and similarly, using Equation 1.2, we have that

$$I\left(X_i, X_{i+2}\right) = -\frac{1}{2}\log\left(1-\rho_{i,i+1}^2\rho_{i+1,i+2}^2\right)$$

Thus, if G_{dir} is triangular, using **??** we should observe the following at the $(i, i+2)$ entry of G_{obs} instead

$$\frac{1}{4}\log\left(1-\rho_{i,i+1}^2\right)\log\left(1-\rho_{i+1,i+2}^2\right)$$

I.e. we make an error (which we could take to be the noise N from **??**) for second order effects equal to

$$-\frac{1}{2}\log\left(1-\rho_{i,i+1}^2\rho_{i+1,i+2}^2\right) - \frac{1}{4}\log\left(1-\rho_{i,i+1}^2\right)\log\left(1-\rho_{i+1,i+2}^2\right)$$

In general, for a Gaussian chain, we have that

$$N_{i,j} = -\frac{1}{2}\log\left(1-\prod_{k=i+1}^{j}\rho_{k-1,k}^2\right) - \left(-\frac{1}{2}\right)^{j-i}\prod_{k=i+1}^{j}\log\left(1-\rho_{k-1,k}^2\right)$$

As we will see in Figure 1.6 and Figure 1.7, for Gaussian chains we can expect some of the same bleeding behavior as observed in Figure 1.4 where we did not use the topological ordering but based the deconvolution on correlation. In particular, from the figures below, we see that for 3-chains, the error is in many cases close to 0 and for most combinations of $\rho_{1,2}$ and $\rho_{2,3}$ less than 0.1. Furthermore, we note that the errors are the largest when it is a strongly connected 3-chain i.e. if both $\rho_{1,2}$ and $\rho_{2,3}$ are close to 1 which again resemble the behavior seen in the case of a symmetrical G_{obs} using correlation as the measure of association although in this case, the error does not propagate to the same extend which we shall also see shortly, when applying the deconvolution algorithm. Notice that as as only the absolute value of the correlation matters, we only show the error for $\rho_{1,2}, \rho_{2,3} \geq 0$.

We extend the above discussion to 4- and 5-chains (i.e. $j = i+3$ and $j = i+4$ in the above expression for N_{ij}) to see how the error propagates in more detail. This is shown in Figure 1.7 for three different scenarios of a 4-chain and a

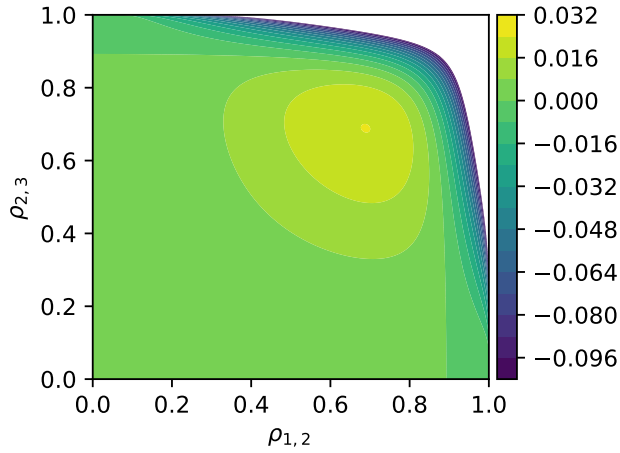
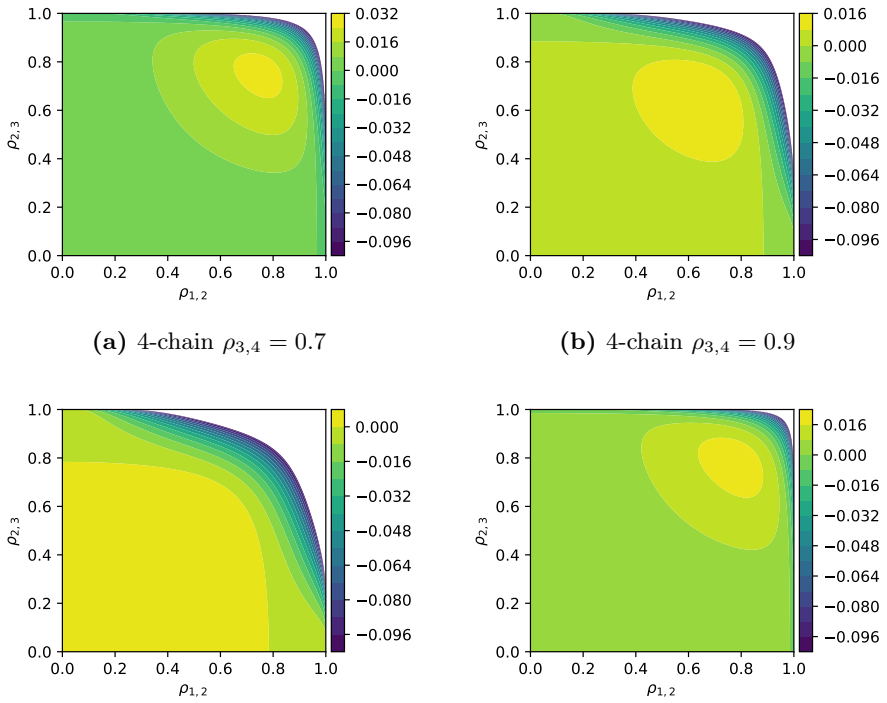


Figure 1.6: The error made by the assumption of G_{obs} and G_{dir} for second order observed effect. Although mutual information does not comply with the underlying assumptions, we observe that in the case of a Gaussian 2-chain, we can expect the error to be relatively small.

single 5-chain. In particular, as the error $N_{i,j}$ is symmetric in $\rho_{1,2}$, $\rho_{2,3}$ and $\rho_{3,4}$ (and $\rho_{4,5}$ in the case of a 5-chain) and because it is hard to accurately show four or five dimensional surfaces, we keep to a fixed set of $\rho_{3,4}$ and $\rho_{4,5}$ when investigating. For the 4-chain, choosing $\rho_{2,3} = 0.9$ (corresponding to mutual information about 0.8304) approximately results in the same error as in Figure 1.6 and if $\rho_{2,3}$ is above e.g. 0.95, we get a worse propagation of errors compared to the 3-chain. Finally, from Figure 1.7d, we see the same picture i.e. that keeping the correlations and hence information between subsequent variable low results in smaller errors in G_{obs} and hence the inferred G_{dir}. Note that under the assumption of a topological ordering such that G_{obs} is strictly upper triangular results in $\rho\left(G_{obs}\right) = 0$ such that no rescaling is necessary (although different choices of the base of the logarithm would have an effect on how much higher order associations influence G_{dir}).

Having obtained a good understanding of how shifting to mutual information instead of correlation in the case of Gaussian chains, we continue with the above example now using mutual information as the elements of G_{obs} based on the correlation matrix from the previous section. Using a triangular G_{obs} we observe similar behavior to that of original example using a triangular G_{obs} but with correlation as can be seen from Figure 1.8. In particular, we do not observe the same magnitude of bleeding effects as in Figure 1.4. However, we observe



(a) 4-chain $\rho_{3,4} = 0.7$

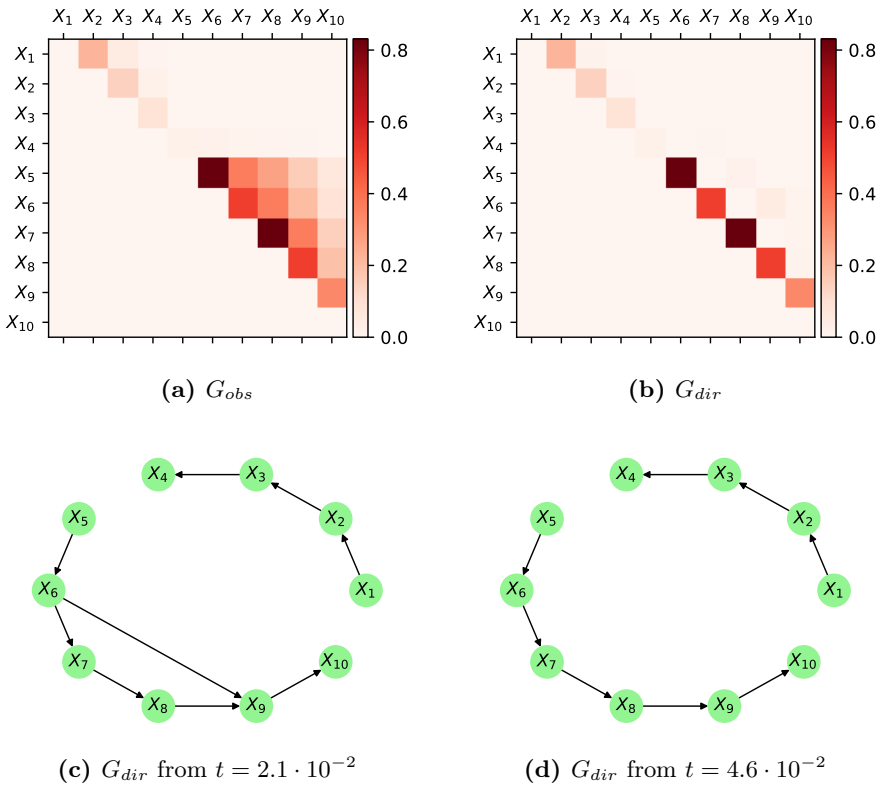
(b) 4-chain $\rho_{3,4} = 0.9$

(c) 4-chain $\rho_{3,4} = 0.95$

(d) 5-chain $\rho_{3,4} = 0.85$, $\rho_{4,5} = 0.6$

Figure 1.7: Errors of convolving mutual information along a 4-chain (a), (b), (c) and a 5-chain (d). Due to symmetry in the expression of the error, only the first 2 links i.e. $\rho_{1,2}$ and $\rho_{2,3}$ are varied on $[0, 1]$ respectively. Only positive correlations are shown as the sign of the correlation cancels in the expression for the error. We note that large correlations and hence large mutual information on each edge results in larger error. In particular, when not too many of the links are strong, we have almost 0 error.

the same tendency to miss weak connections as was also observed in Figure 1.4. All in all, we get very good results using a triangular G_{obs} even though mutual information does not have the same properties as correlation. In particular, this is what we expected as we have only used $\rho_{i,i+1} \leq 0.9$, from the above investigation of the error.



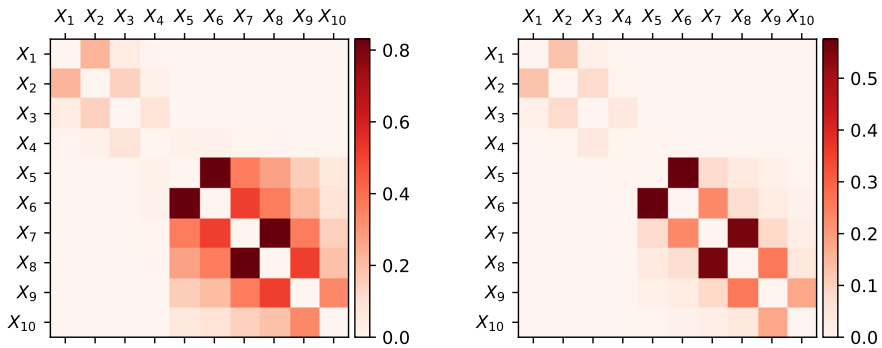
(a) G_{obs} **(b)** G_{dir}

(c) G_{dir} from $t = 2.1 \cdot 10^{-2}$ **(d)** G_{dir} from $t = 4.6 \cdot 10^{-2}$

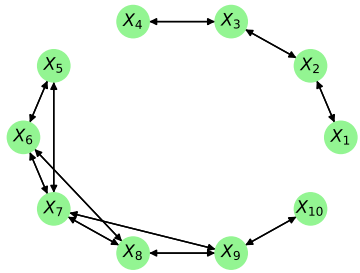
Figure 1.8: Using mutual information as the measure of similarity as well as assuming a topological order i.e. making G_{obs} strictly triangular as seen in (a) we almost perfectly infer G_{dir} as seen in (b) except for $[G_{dir}]_{6,9}$. Choosing cutoffs $t = 2.1 \cdot 10^{-2}$ (c) and $t = 4.6 \cdot 10^{-2}$ (d) it is clear that adjusting the threshold we can get a better result than using a symmetric G_{obs} with correlation.

Finally, we use the corresponding symmetric G_{obs} (rescaled such that the largest absolute eigenvalue of G_{dir} is 0.99) which results in G_{dir} and the graph using a threshold $t = 4.88 \cdot 10^{-2}$ shown in Figure 1.9. Again, we observe some bleeding on the more strongly connected sub-chain as with the symmetric G_{obs} using correlation in Figure 1.4. Again, we observe comparable results and note that increasing the threshold would disconnect X_3 and X_4 before removing the higher order effects.

In conclusion, we have seen what errors can arise in the discovered network using both correlation and mutual information as the measure of association. Namely, long strongly connected chains seem to be a problem if one does not know a



(a) G_{obs} **(b)** G_{dir}



(c) G_{dir} as a graph

Figure 1.9: Using a symmetric G_{obs} containing the observed mutual information (a) we infer a G_{dir} (b) comparable to that if we had used correlation instead. Choosing the threshold $t = 4.88{\cdot}10^{-2}$ seem a good compromise between connectedness and density resulting in an almost identical discovered network structure to that of using a symmetric correlation G_{obs}.

topological ordering of the variables, in which case these are heavily reduced as seen in Figure 1.3 and Figure 1.8. Thus, we proceed in the next section by considering a more complicated underlying (Gaussian) network to observe if other unwanted effects can occur and if a topological ordering is necessary if the network is not simply a path.

1.2 Directed acyclic Gaussian graphs

In this section, we will expand on the results from the previous section by considering a more general structure. In particular, let $\mathcal{G}$ be a directed acyclic graph with nodes corresponding to variables from a random vector $\boldsymbol{X}$ with directed edges indicating direct dependencies. Clearly, such a DAG has a topological ordering and as such we shall index the variables 1 through d such that if the index of a variable is i, and j is the index of another element of the random vector $\boldsymbol{X}$, then $i < j$ implies there is no (directed) path from j to i. Note that since a topological ordering is not necessarily unique, we can not infer that there is a (directed) path from i to j or even if k is reachable form j (i.e. there exists a path from j to k) it does not follow that k is reachable from i. In Figure 1.10 a subset of such a DAG is shown with a possible labelling where $i < j$ and $k_m < k_n$ when $m < n$. It is then the weights along these directed edges which we will once again call G_{dir} that we wish to infer based on the transitive closure. As an example, from Figure 1.10, the transitive closure would result in an observed similarity between i and j although no 1 path i.e. single direct edge connects the two variables. From the definition of the labels, it is clear

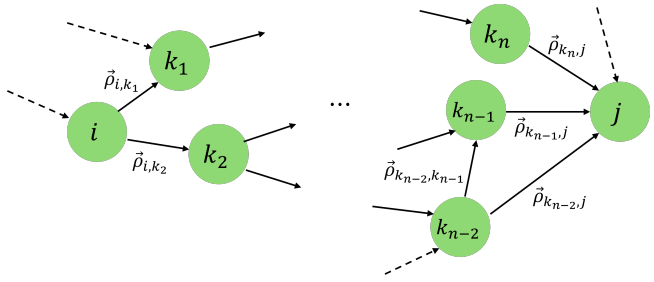


Figure 1.10: A general (linear) network represented as a DAG. The directed edge weights $\vec{\rho}_{k,l}$ specify how much the variable index k make up of variable l. Although i and j are not directly connected, multiple paths may exist between the two nodes, making the propagation of similarity more complex from that of a chain.

that G_{dir} is once again strictly upper triangular as entries below the diagonal corresponds to edges going from a random variable with an index i to another random variable with index j such that $i > j$ which is clearly a contradiction. Also, the diagonal elements are 0 as there can not be any loops in DAGs.

Similarly to the definition of (Gaussian) chains, based on d independent (or even just pairwise uncorrelated) random variables Z_i we can define a general network

of random variables X_i based on $\boldsymbol{Z}$ in the following way

$$\begin{aligned} X_1 &= Z_1 \\ X_2 &= \vec{\rho}_{1,2} X_1 + \sqrt{1 - \vec{\rho}_{1,2}^2} Z_2 \\ X_3 &= \vec{\rho}_{1,3} X_1 + \vec{\rho}_{2,3} X_2 + c_3 Z_3 \\ &\vdots \\ X_d &= \sum_{k<d} \vec{\rho}_{k,d} X_k + c_d Z_d \end{aligned} \tag{1.4}$$

where c_i is chosen such that $\mathrm{Var}(X_i) = 1$ to make the analysis later on carries out simpler as then $\vec{\rho}_{i,j}$ is actually the *direct* correlation between the variables indexed i and j as shown in Figure 1.10. Of course, for the variance of each random variable to be 1 there must be some constraints on the chosen $\vec{\rho}_{i,j}$ such as neither one of them can exceed 1 in absolute value. Furthermore, consider the following bound on the variance of X_i assuming c_k for $k < i$ have been chosen such that $\mathrm{Var}(X_k) = 1$.

$$\begin{aligned} \mathrm{Var}[X_i] &= \sum_{k<i} \vec{\rho}_{k,i}^2 + 2 \sum_{k<l<i} \vec{\rho}_{k,i} \vec{\rho}_{l,i} \mathrm{Cov}[X_k, X_l] + c_i^2 \\ &\leq \sum_{k<i} \vec{\rho}_{k,i}^2 + 2 \sum_{k<l<i} \vec{\rho}_{k,i} \vec{\rho}_{l,i} + c_i^2 \\ &= \left(\sum_{k<i} \rho_{k,i} \right)^2 + c_i^2 \end{aligned} \tag{1.5}$$

where we have used that Z_i is uncorrelated with X_k for $k < i$ and that the covariance between variables with variance 1 is at most 1 to obtain the inequality. Hence, choosing the sum of the ingoing edges to be at most 1 for every node ensures that the constants c_i for $i \in \{2, \ldots, d\}$ exist in order to make the variance of each X_i 1. This, we will use in the following example to easily build a network such that $\vec{\rho}_{i,j}$ is the pure correlation.

However, before constructing an example and using bot correlation and mutual information we must determine the theoretical G_{obs} for both cases. To do this, we shall consider the (i, j) element of G_{obs} when using correlation as a measure of similarity and later use mutual information based on these correlations and Proposition 1.1 in the case of $\boldsymbol{Z}$ being a Gaussian random vector. To calculate $[G_{obs}]_{i,j}$ we shall consider the immediate predecessors to node j in the graph $\mathcal{G}$ corresponding to Equation 1.4. The immediate predecessors or *in-neighbors* of a node j is denoted $N^-(X_j)$ or in shorthand notation N_j^-. An example of this is shown in Figure 1.11 where the in-neighbors of j has been marked in red. With this notation, we proceed with the computation of the (i, j) entry of G_{obs} which is the covariance between X_i and X_j when $i < j$ and 0 elsewhere.

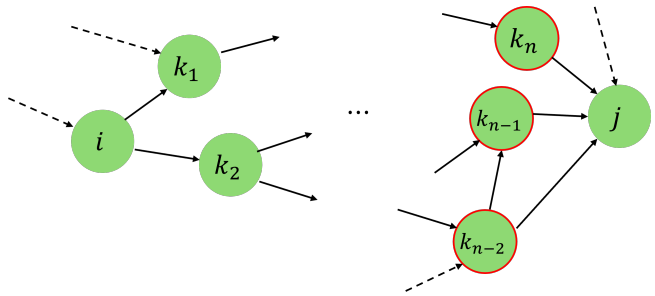


Figure 1.11: For node j, the set N_j^- is illustrated with red borders. It is exactly the set of nodes going directly into j. We note that an in-neighbor l of in-neighbor k of node j can also be an in-neighbor of j i.e. l can influence both k and j whilst k also directly influenced j. It is in particular these direct dependencies we want to be sure of as their existence makes the network more complex but failing to discover these can lead to a significant reduction in prediction accuracy.

$$\begin{aligned}
[G_{obs}]_{i,j} &= \mathrm{Cov}\left[X_i, \sum_{k\in N_j^-} \vec{\rho}_{k,j}\, X_k + c_j Z_j\right] \\
&= \mathrm{Cov}\left[X_i, \sum_{k\in N_j^-} \vec{\rho}_{k,j}\, X_k\right] \\
&= \sum_{k\in N_j^-} \vec{\rho}_{k,j} \mathrm{Cov}\left[X_i, X_k\right] \\
&= \sum_{k\in 1}^{j-1} \vec{\rho}_{k,j} \mathrm{Cov}\left[X_i, X_k\right] \\
&= \vec{\rho}_{i,j} + \sum_{k\in 1}^{d} \vec{\rho}_{k,j}\, [G_{obs}]_{i,k}
\end{aligned} \tag{1.6}$$

For the fourth equality, we have used that $\vec{\rho}_{k,j} = 0$ whenever $k \notin N_j^-$ which again for the fifth equality holds for any $k \geq j$. Furthermore, since $[G_{obs}]_{i,i} = 0$ we need to add $\vec{\rho}_{i,j}$ to make the final equality hold. It is clear that the above can also be expressed as a matrix equation, namely

$$G_{obs} = G_{obs} G_{dir} + G_{dir}$$

Hence, as G_{dir} is strictly upper triangular thus making $I - G_{dir}$ invertible, we can directly express G_{obs} in terms of G_{dir}. We find that

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

which we recognize as **??** hence also for a general network (and not just a chain), using correlation and knowing/assuming a topological order of the random variables we are able to perfectly rediscover G_{dir} from G_{obs}.

With the above, we then define an example Gaussian network with the following weights and shown in Figure 1.12 to get a better understanding of this example hopefully should reappear after deconvolution using both correlation and mutual information respectively.

$$\begin{aligned} \vec{\rho}_{1,2} &= 0.7, \quad \vec{\rho}_{5,6} = 0.5, \quad \vec{\rho}_{2,7} = 0.3 \\ \vec{\rho}_{6,7} &= 0.3, \quad \vec{\rho}_{6,8} = 0.7, \quad \vec{\rho}_{4,9} = 0.3 \\ \vec{\rho}_{8,9} &= 0.3, \quad \vec{\rho}_{7,10} = 0.4, \quad \vec{\rho}_{9,10} = 0.2 \end{aligned} \tag{1.7}$$

In particular, from Equation 1.5 and Figure 1.6 and Figure 1.7, we suspect that the bleeding effects observed for the Gaussian chain won't appear to the same extent in this case.

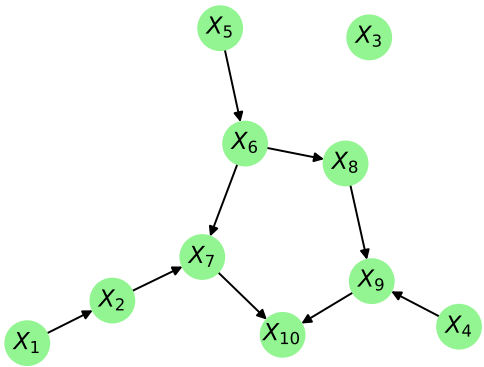
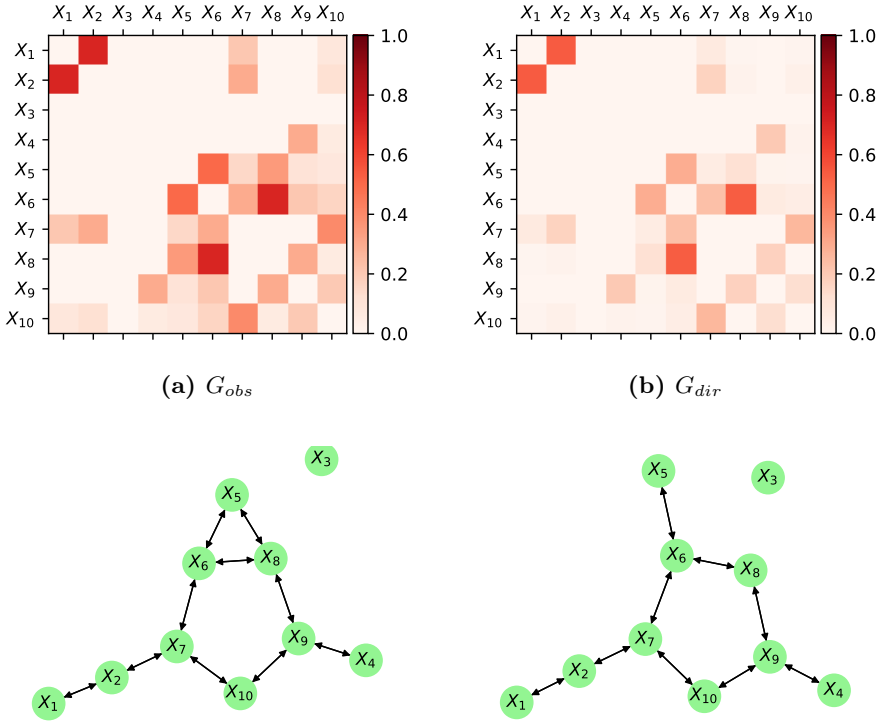


Figure 1.12: The graph defined in Equation 1.7. Note that X_3 is neither influenced nor influences any other variable. It is of course in our interest to accurately tell if X_3 should be considered if we try to infer a probability distribution on e.g. X_{10} given observations of the other variables.

Applying the deconvolution algorithm, we obtain the results in **??** which trivially, from the above analysis on G_{obs}, results in a perfect reconstruction of the network. If instead, we do not assume a topological structure, we can also recover the structure, although we need to tune the threshold as can be seen from Figure 1.13. Tuning the α and β did not have much of an effect. Actually, decreasing β seemed to worsen the results which is also in line with our expectations as choosing smaller β skews the effects of higher order interactions. Thus,

it is primarily the threshold that we want to tune in this case and choosing $t = 1.18 \cdot 10^{-1}$ we accurately infer the network structure contrary to the results from the Gaussian chain. However, we still observe second order effects i.e. the edge between X_5 and X_8 which was also the case in Figure 1.4 Finally, before



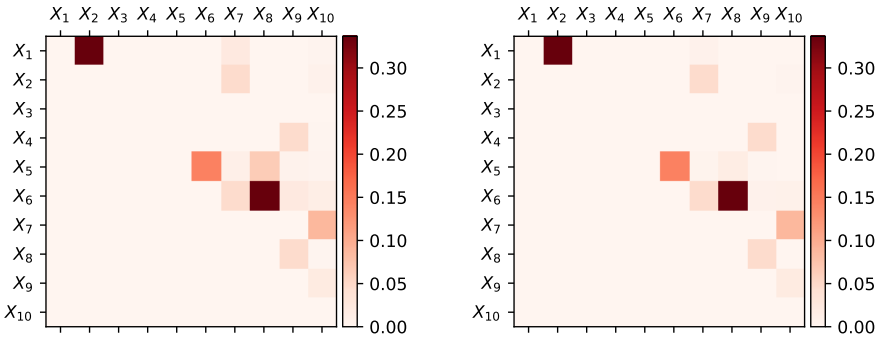
(a) G_{obs} **(b)** G_{dir}

(c) G_{dir} from $t = 6.4 \cdot 10^{-2}$ **(d)** G_{dir} from $t = 1.18 \cdot 10^{-1}$

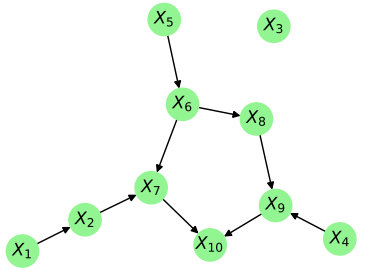
Figure 1.13: Not knowing the topological structure and thus using a symmetric G_{obs} (a) we obtain the G_{dir} in (b). Clearly, there is some bleeding, but choosing the threshold $t = 1.18 \cdot 10^{-1}$ we can accurately rediscover the network structure up to a direction on the edges. As with the previous example of Gaussian chains, we observe some tendency to inaccurately filter out second order effects as can be seen in (c) where X_5 and X_8 is connected.

continuing with results regarding the different methods for estimating mutual information, we present the results from above using mutual information instead of correlation as the measure of similarity. Namely, once again assuming the topological order such that G_{obs} is strictly upper triangular and hence no need for rescaling we get the results shown in Figure 1.14. As expected, we observe on par performance to using correlation. Only the edge from 5 to 8

being almost as strong as 9 to 10 could be a problem i.e. choosing a threshold a little larger than $t = 1.7 \cdot 10^{-2}$ (which is quite small and has been used for Figure 1.14c) would have resulted in an edge from X_5 to X_8. Hence, in a real world example we might have chosen to either leave out both edges which depending on the scenario may or may not be an acceptable error or include the both of them.



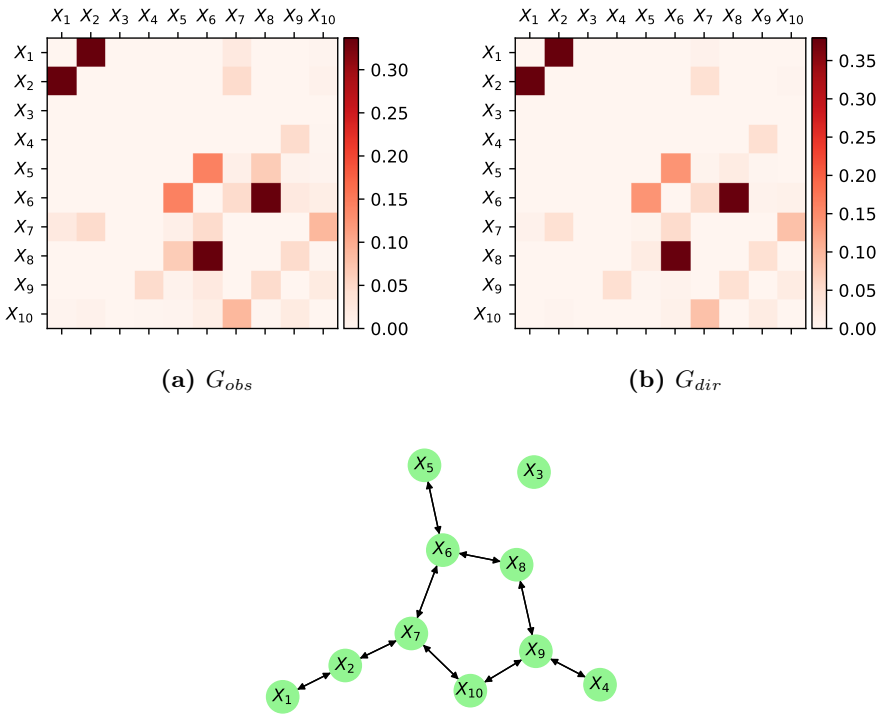
(a) G_{obs} **(b)** G_{dir}



(c) G_{dir} as a graph

Figure 1.14: Using mutual information instead of correlation results in G_{obs} shown in (a). The non-linear map from correlation to mutual information only effects the resulting G_{dir} a little as shown in (b) when comparing to the $\vec{\rho}_{i,j}$ from Equation 1.7. Choosing the relatively small threshold $t = 1.7 \cdot 10^{-2}$ results in a perfect reconstruction of the graph structure.

Furthermore, using a symmetric G_{obs} instead i.e. no assumption on topology, does not seem to have much of an effect as seen from Figure 1.15. Although there still is a small weight on the edge from X_5 to X_8, by choosing the threshold $t = 1.96 \cdot 10^{-2}$ we can accurately construct the true network structure.



(a) G_{obs} **(b)** G_{dir}

(c) G_{dir} as a graph

Figure 1.15: Using a symmetric G_{obs} instead of an upper triangular G_{obs} result in near identical G_{dir} in terms of relative weights on the edges. Namely, the G_{dir} shown in (b) seem to be almost a scaled version of the (reflected) G_{dir} derived from a triangular G_{obs}. Thus, as (c) also shows, we can accurately infer the structure of the network using a threshold $t = 1.96 \cdot 10^{-2}$.

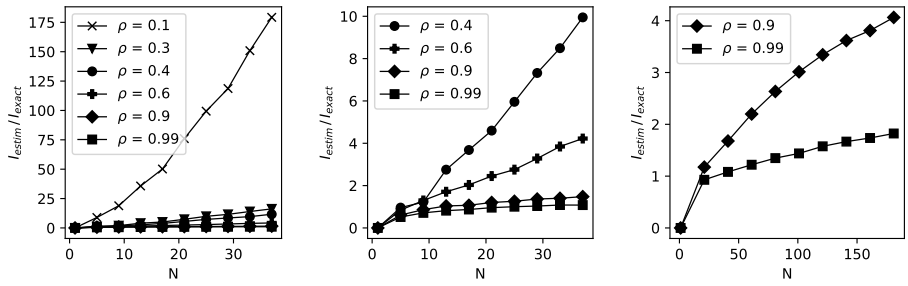
In conclusion, we observe a useful property of more general networks that for both mutual information and correlation, the additional assumption of the topological order does not have much of an effect in these cases contrary to what we observed for Gaussian chains and linear chain models in general, when using correlation.

1.3 CE computation

Having discussed the strengths and weaknesses of **??**, we now turn our attention to **??**. Namely, in this section we shall discuss the different methods from **??** and how they perform on two examples. Once again, we shall base our results on two examples. The first is a simple case, where we shall see what to be aware of when initially the observations are transformed through estimated distribution functions as well as how accurate the different methods for estimating the Copula entropy i.e. mutual are. Continuing from the first example, we shall once again consider the network from Section 1.2 specified by Equation 1.7. In particular, we will see how well the combined framework performs on an example we have already seen to be quite solvable if one uses accurate estimates of the mutual information which we previously calculated theoretically.

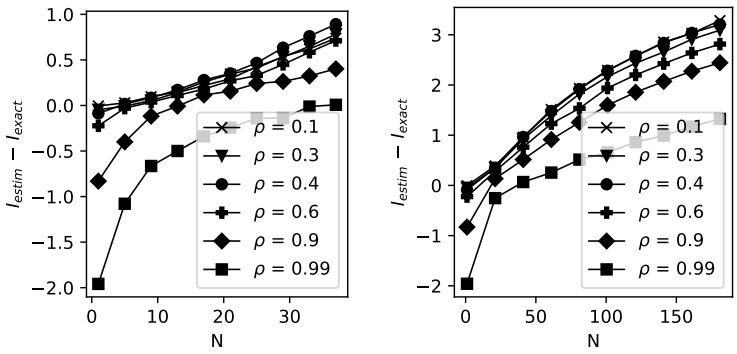
1.3.1 Spline and KDE based CE estimation

Before discussing the first example, we shall however discuss the problem with the spline based method and using histograms in general. Namely, we shall first see that if one were to just simply us a raw binning approach, the number of bins N influences the estimate a lot and no number of bins seems to perform well on all cases. Namely, let $\boldsymbol{X}$ be a bivariate Gaussian with correlation ρ, then the Copula density looks as in Figure 1.30. In particular, we notice the peaks at $(0,0)$ and $(1,1)$ which result in a lot of mutual information. Now, simulating $n = 400$ observations from the joint distribution and transforming to the unit square through the marginal distribution function for varying correlations ρ, we can compare the estimated mutual information I_{estim} using the results from **??** to the true mutual information given by $I_{exact} = -\frac{1}{2}\log\left(1-\rho^2\right)$. The results are shown in Figure 1.16 where we report the relative size of the estimate and the exact mutual information and in Figure 1.17 where the difference is reported. From Figure 1.16, we might choose $N \approx 10$ as in [**?**] however for large correlations, we drastically underestimate mutual information. Increasing the number of bins to e.g. $N = 50$ corrects this error for large correlations, but then small correlations have a relatively large error of around 0.5 which is a relatively large error when comparing to the error the deconvolution step makes as illustrated in Figure 1.7.



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

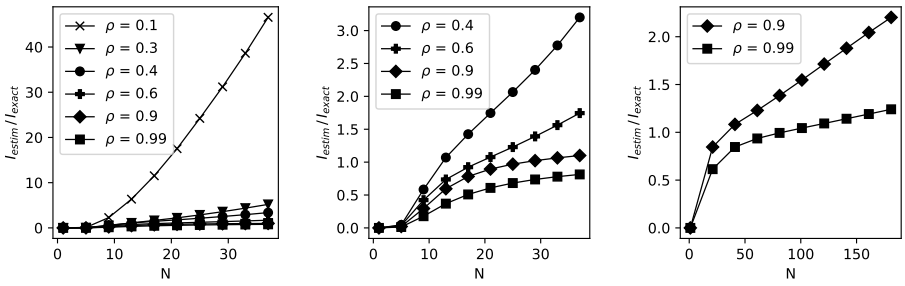
Figure 1.16



(a) **(b)**

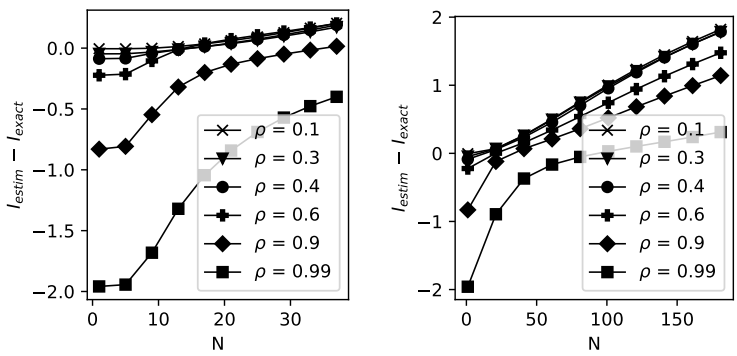
Figure 1.17

We thus proceed with using the B-spline approach. Similar to the above results, in Figure 1.18 and Figure 1.19, we observe that B-spline approach is prone to the same errors as the raw binning approach. However, from Figure 1.19 we see that the error are smaller for the B-spline based approach but also that a better choice for the number of bins is around $N = 60$ contrary to the results of [**?**].



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

Figure 1.18: 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)



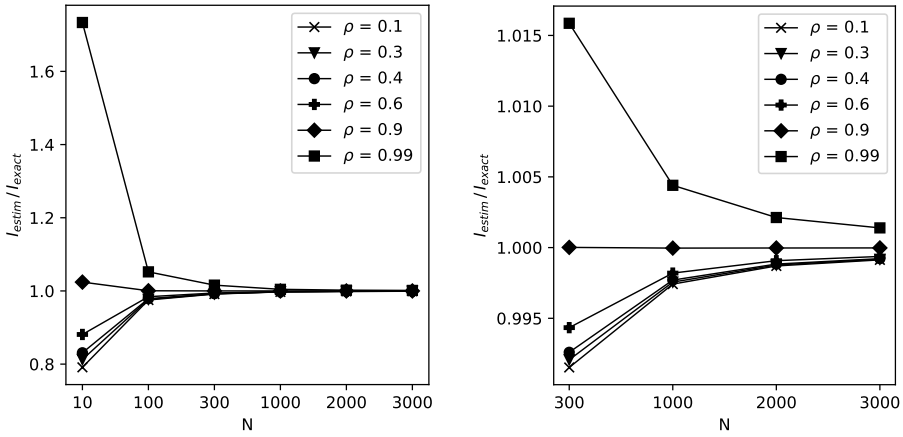
(a) **(b)**

Figure 1.19: through repeated trials, we see that this method is actually quite robust when varying the number of observations n

The results for M-splines are shown in **??** and **??** and we observe comparable performance except perhaps for a better estimation of mutual information for large correlations which is what we would expect from our discussion in **??**.

The problem we observe with the above methods is that when mutual informa-

tion is large, the Copula density is close to a one-dimensional manifold. Hence, to calculate the mutual information well, need many bins. However, with many bins the estimate becomes more noisy as the support of each spline shrinks with $\frac{1}{N}$. However, as seen from Figure 1.20, if we can accurately estimate the Copula density function from observations, we can compute the mutual information perfectly by increasing the fineness of the integral approximation. In particular, the results below where obtained through the theoretical Copula density function evaluated at the bin centers $\left(\frac{2i-1}{2N}, \frac{2j-1}{2N}\right)$ for $i, j \in \{1, \ldots, N\}$. We see that this simple approximation with $N = 1000$ is good for correlations up to $\rho = 0.99$. However, due to numerical limitations, we shall use $N = 500$ and only 400 observations to evaluate the performance of the KDE from the previous chapter. We note that a more efficient implementation is possible splitting up the computation in multiple parts as the problem with many observations and bins is to compute the joint density function which is of $\mathcal{O}\left(N^2 n\right)$



(a) **(b)**

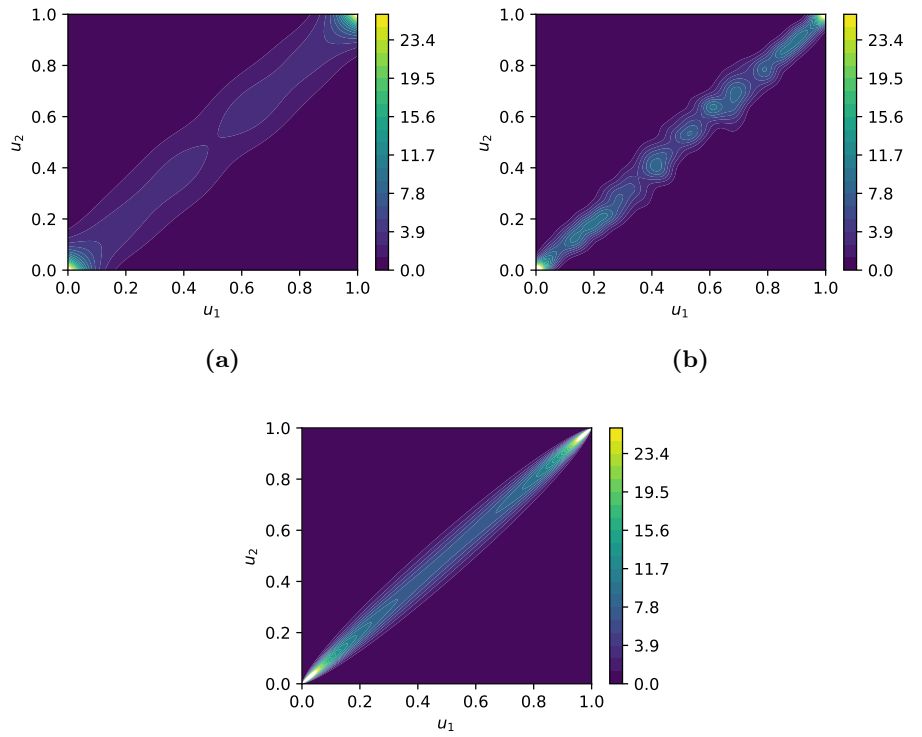
Figure 1.20: Evaluation of MI for new method for different N. Bør sammenlignes med artiekl fundet (har sat i bibtex) og original papers (ikke Kina)

the chosen h is around 0.085 every time using Scott's rule. As we see from Table 1.1, we have relatively low variance and in general, we compute the mutual information to a higher accuracy than either B-splines and M-splines. Thus, in the following example, we will only consider this method for estimating the mutual information between pairs of variables. In Figure 1.21 we have shown the estimated density and the theoretical copula. We observe that indeed the method accurately estimates the Copula density, although we note that the concept of a local bandwidth as discussed in **??** is likely to improve on the results

ρ	h	mean error	variance
0.1	h^{Scott}	0.01583	$5.3446 \cdot 10^{-5}$
0.3	h^{Scott}	0.02302	$3.7957 \cdot 10^{-4}$
0.4	h^{Scott}	0.006898	$3.2655 \cdot 10^{-4}$
0.6	h^{Scott}	0.007803	$1.0027 \cdot 10^{-3}$
0.9	h^{Scott}	-0.1844	$4.4478 \cdot 10^{-4}$
0.99	h^{Scott}	-1.007	$1.6328 \cdot 10^{-4}$
0.99	$0.3\, h^{Scott}$	-0.3468	$1.0616 \cdot 10^{-3}$

Table 1.1: based on 10 trials

as peaks at $(0, 0)$ and $(1, 1)$ does not quite resemble those of the theoretical Copula density. In particular, we observe that reducing the bandwidth improves on the estimate and is clearly observe by the improved resemblance with the theoretical Copula density. Although this is at the cost of undersmoothing on the interior. Indeed, a K-means based estimator of the bandwidth h could work well as the mean distance near $(0, 0)$ and $(1, 1)$ is very small compared to the interior. However, we note that as long as observations are not almost on a line, the KDE performs quite well. This, we shall also see in the next section where we couple the above discussions on mutual information estimation with the deconvolution algorithm.



(a) **(b)**

(c)

Figure 1.21

1.3.2 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

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0.9\sigma_1\sigma_2 & 0 \\ 0.9\sigma_1\sigma_2 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix} = \operatorname{diag}(\boldsymbol{\sigma}) \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \operatorname{diag}(\boldsymbol{\sigma})$$

In particular, in terms of Equation 1.4, we have that for $\boldsymbol{X}$, $\vec{\rho}_{1,2} = 0.9$. It is clear that to **??**, the mean of $\boldsymbol{X}$ is of no importance as it simply corresponds to a scaling of the Y_i variables. Furthermore, because of **??**, 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 **??**. Additionally, different $\boldsymbol{\sigma}$ corresponds to different scaling of $\boldsymbol{X}$, and thus we should observe equal or near equal G_{dir} for all $\boldsymbol{Y}$ independently of $\boldsymbol{\sigma}$. Initially, we shall see how this hypothesis holds

up when considering the following three examples

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

To draw from this distribution, one can either use built-in functions or use the Cholesky factorization of the correlation matrix to generate proper correlated variables from 3 independent standard normal distributions and then scale with the chosen standard deviation to generate samples from all three cases based on the same seed. We shall do the latter and also generate a generous number of samples (10,000) such that the KDE based methods have the best possible prerequisites whilst also being numerically tractable later on.

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 $\boldsymbol{\sigma} = (1, 1, 1)$, **??** and **??** returns the following (using $\alpha = 1$ and $\beta = 0.99$)

$$G_{dir} = \begin{bmatrix} -0.3363 & 0.6552 & 0.08813 \\ 0.6552 & -0.3328 & 0.06480 \\ 0.08813 & 0.06480 & -0.01734 \end{bmatrix} \tag{1.8}$$

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

$$G_{dir} = \begin{bmatrix} -0.3289 & 0.6610 & 0.004827 \\ 0.6610 & -0.29889 & 0.004977 \\ 0.004827 & 0.004977 & -0.00007197 \end{bmatrix} \tag{1.9}$$

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

$$G_{dir} = \begin{bmatrix} -0.1836 & 0.3117 & 0.2252 \\ 0.3117 & -0.4065 & 0.5962 \\ 0.2252 & 0.5962 & -0.3673 \end{bmatrix}$$

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 **??** 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 **??** we immediately see a flaw in the supposedly uniform variables U_i as shown in figure Figure 1.22

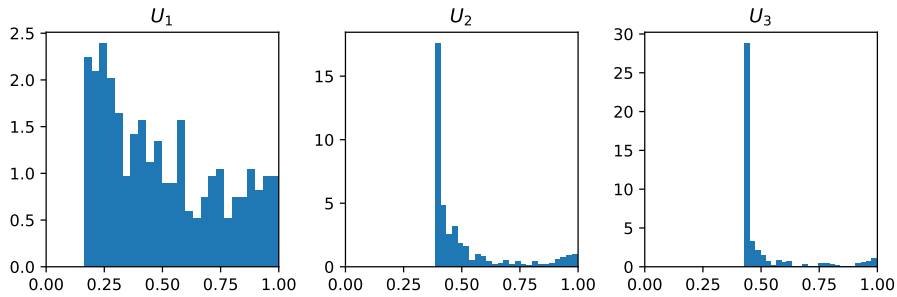


Figure 1.22: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 2, 3)$. These should be uniformly distributed, but clearly this is not the case for U_2 and U_3. Even U_1 does not quite resemble 10,000 samples from a uniform distribution.

	U_1	U_2	U_3
D_n	0.16512	0.38354	0.42764
p-value	0	0	0

Table 1.2: based on 10,000 samples for $\boldsymbol{\sigma} = (1, 2, 3)$.

Before handling this, the non-uniformity of U_1 in Figure 1.22 is likely also present in the case when $\boldsymbol{\sigma} = (1, 1, 1)$. Indeed, Figure 1.23 shows that this is indeed the case.

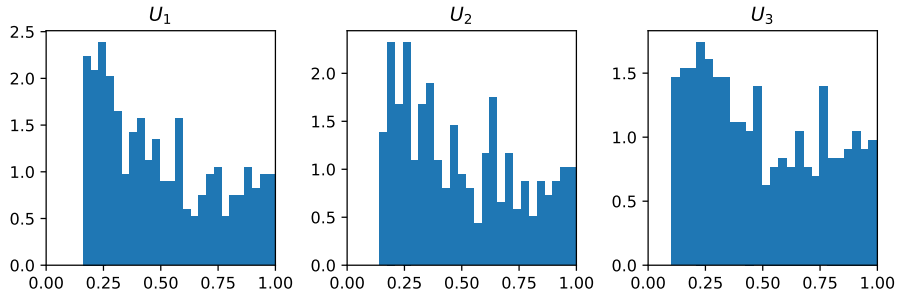


Figure 1.23: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 1, 1)$.

	U_1	U_2	U_3
D_n	0.16511	0.14672	0.10561
p-value	0	0	$2.3819975157518748 \cdot 10^{-4}$

Table 1.3: based on 400 samples for $\boldsymbol{\sigma} = (1, 1, 1)$.

Finally, just to be sure, $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ is also shown in Figure 1.24 and seems very reasonable, except for U_3.

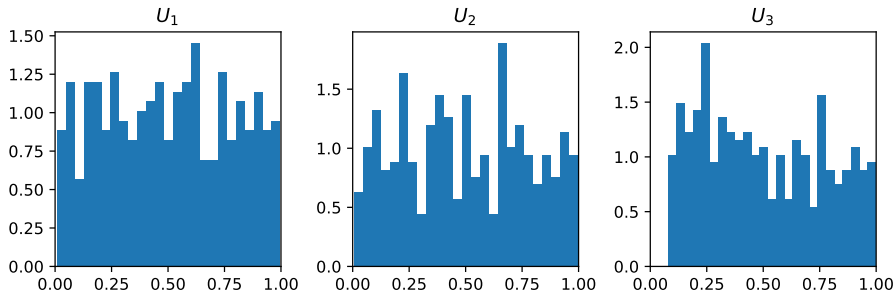


Figure 1.24: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

	U_1	U_2	U_3
D_n	0.029036	0.029026	0.085611
p-value	0.88427	0.88454	0.0052791

Table 1.4: based on 10,000 samples 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 1.25 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.

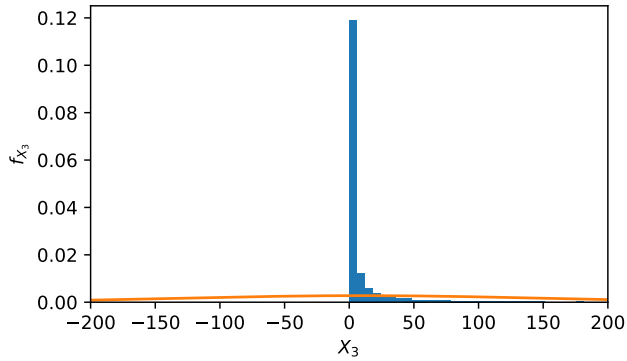


Figure 1.25

The poor fit also explains the high concentration of U_3 around 0.5 in Figure 1.22 as only 54.5% of the probability mass lies above 0.

However, also here **??** 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 **??** and the resulting $u_i^{(j)}$ is shown in Figure 1.26.

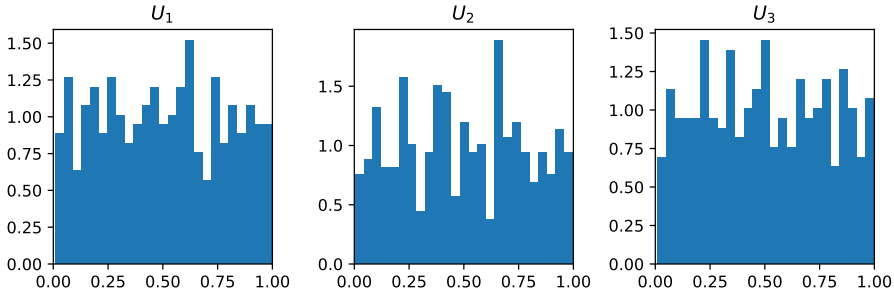


Figure 1.26

The resulting $u_i^{(j)}$ now seem to follow a uniform distribution and indeed the KDE fits much better as seen in Figure 1.27.

	U_1	U_2	U_3
D_n	0.0061099	0.0061435	0.0073148
p-value	0.84838	0.84368	0.65690

Table 1.5: based on 10,000 samples for $\boldsymbol{\sigma} = (1, 2, 3)$ with power transform.

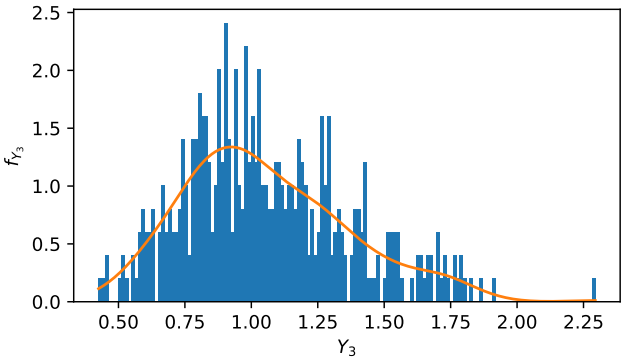
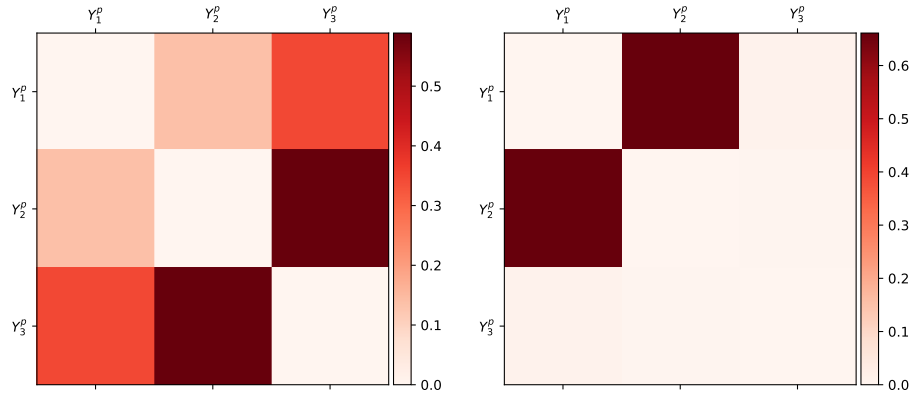


Figure 1.27

Turning to **??** and **??** we now find that G_{dir} is given by

$$G_{dir} = \begin{bmatrix} -0.3290 & 0.6610 & 0.01188 \\ 0.6610 & -0.3289 & 0.004603 \\ 0.01188 & 0.004603 & -0.0002167 \end{bmatrix}$$

Which is indeed much more comparable with the result from before in Equation 1.8 and Equation 1.9. The difference between G_{dir} from $\boldsymbol{Y}$ and $\boldsymbol{Y}^p$ is clearly visible in Figure 1.28 and also Figure 1.28b resembles the original correlation structure.



(a) **(b)**

Figure 1.28: G_{dir} resulting from 400 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 **??** can also be calculated theoretically. For this, we shall use Proposition 1.1 which permits a theoretical result, namely

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

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

$$G_{obs} = \begin{bmatrix} 0. & 0.64069542 & 0.01824538 \\ 0.64069542 & 0. & 0.0135368 \\ 0.01824538 & 0.0135368 & 0. \end{bmatrix}$$

Test om denne G er lige den teoretiske. Eller nærmere, argumenter for hvorfor vi ikke laver en test, eller hvad man kunne gøre. Har samplet fra en simultan normalfordeling, så kan lave en til en mellem MI og korrelation.

From the confidence density for the correlation ρ given the emperical correlation

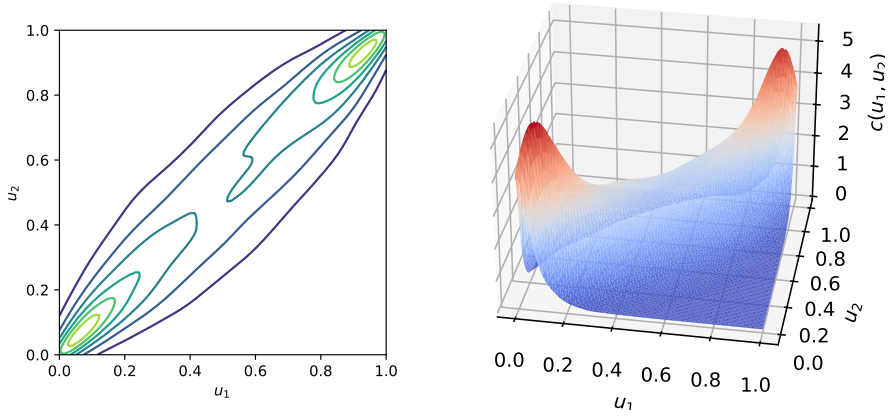
r is given by

$$f(\rho \mid r, \nu) = \frac{\nu(\nu-1)\Gamma(\nu-1)}{\sqrt{2\pi}\Gamma(\nu+\frac{1}{2})} \frac{\left(1-r^2\right)^{\frac{\nu-1}{2}} \left(1-\rho^2\right)^{\frac{\nu-2}{2}}}{(1-r\rho)^{\frac{2\nu-1}{2}}} F\left(\frac{3}{2}, -\frac{1}{2}, \nu+\frac{1}{2}, \frac{1+r\rho}{2}\right)$$

from the mutual information, we can calculate the absolute correlation. Notice that the density does not change when reversing both r and ρ simultaneously, thus, without loss of generality, assume $r \geq 0$, then we can calculate a CI for ρ (which will be negated if we had used $-r$ instead and thus would be identical when taking the absolute value). If the original CI $[a, b]$ contains 0 i.e. $a < 0$, we shall write the CI for the absolute correlation as $[0, b]$ instead. This way, we can compare the absolute correlations and see if they are the same (by checking if the CI contains the theoretical correlation) by [**?**]. Using numerical integration (fast enough with high numerical accuracy from many bins, 1 mil bins, yielding probability mass 1.0000000000008133), can compute CI for absolute correlation

??

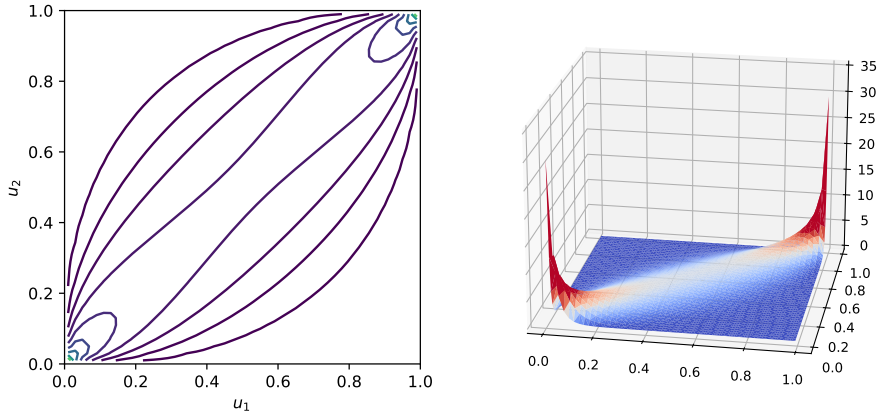
Clearly these are not equal, but in this case, the error is suspected to originate from the estimated joint density. For example, considering X_1 and X_2, we compare the estimated joint copula density and compare to the theoretical reference til et sted hvor gausisk copula står shown in Figure 1.29 and Figure 1.30 respectively.



(a) **(b)**

Figure 1.29: 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 **??** 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 1.30: 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 1.1. *From Subsection 1.3.2 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*

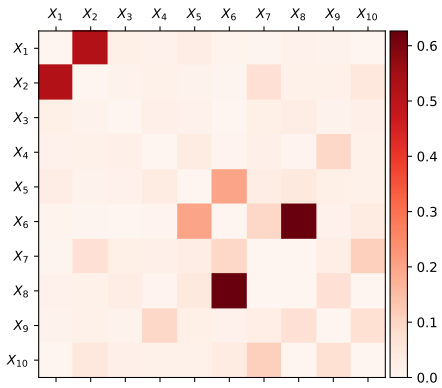
1.3.3 10D gaussian example

None are significant (marginal distribution).

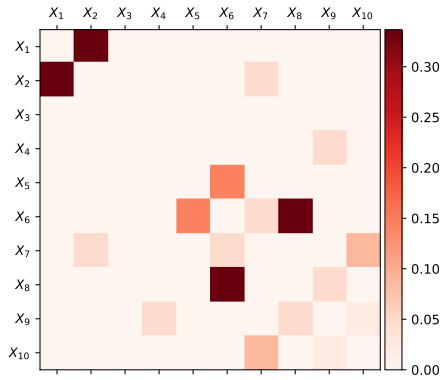
Also, we expect this to do well, since there are no correlations above 0.7. Thus, from Table 1.1 the errors in the mutual information estimates should be quite small.

the setup is like before that we simulate 400 observations from the network, defined by Equation 1.7

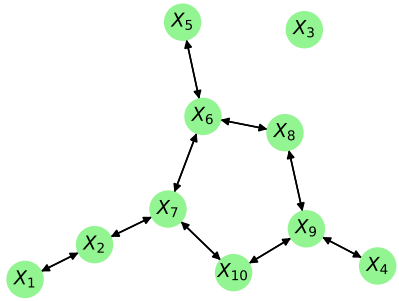
casuality svarer til at lave nedre/øvre trekant. Er der forskel i at gør edet før og efter for en symmetrisk matrix? - Ja, men begge metoder på 10 eksempel giver gode resultater. Kommenter at det er matematisk meget forskelligt at filtrere først og så ND efter og omvendt



(a) $\hat{G}_{dir}$

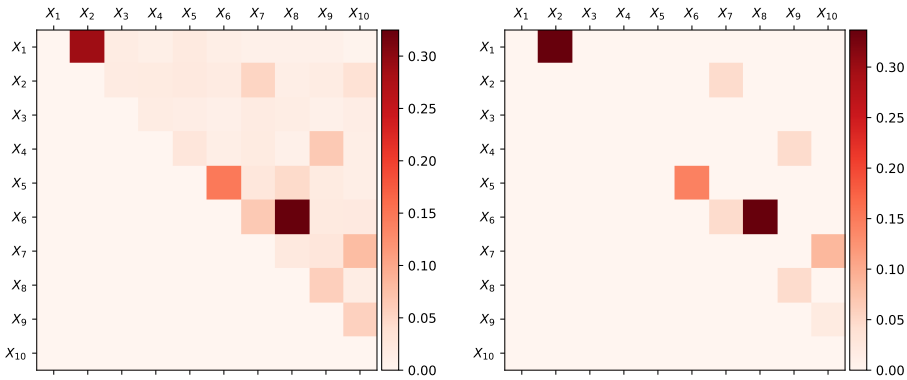


(b) G_{dir}

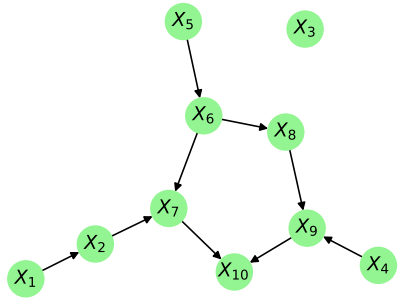


(c) $\hat{G}_{dir}$ as a graph

Figure 1.31: Using a symmetric G_{obs}



(a) $\hat{G}_{dir}$ **(b)** G_{dir}



(c) $\hat{G}_{dir}$ as a digraph

Figure 1.32: Using a triangular G_{obs}