Chapter 1

Results and Discussion

In this section, we will apply **??** and **??** on different examples to assess the capabilities and pain points of the framework. In particular, we shall initially consider a simple network whose causal structure is a simple chain and where only linear correlation exists between the random variables constituting the chain. We shall see that using correlations and an assumption of the topological structure, we can perfectly reconstruct the causal structure from G_{obs}. Furthermore, we shall see that using mutual information as the measure of association instead, we are still able to reconstruct the causal structure. We also argue why mutual information is preferable in a general setting and quantify the resulting errors.

After having considered a simple causal structure, we extend to a general network, with variables still only linearly correlated. Once again, using correlation and an assumption of the topological order of the random variables, we can perfectly reconstruct the causal structure. We shall however also use mutual information and observe that once again, we infer the true causal structure.

After the above results concerning **??**, we shall compare the different methods from **??** for estimating mutual information. In particular, we shall see how the estimator used in [**?**] performs on a few samples and compare this to the KDE-based method.

From this, we shall investigate how the algorithms work when combined. Specifically, we shall observe that preconditioning of the variables is important for accurate estimates of mutual information. Furthermore, we shall see that based on only a few hundred samples from the above network, the causal structure is recovered.

Finally, we shall use the combined learnings from the above on the pharmaceutical dataset, described in **??**. In particular, depending on the included variables and assumptions of topological structure, we shall infer a few possible causal structures and comment on these.

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 use a multivariate Gaussian 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 are independent of the mean and variance of each of the variables however for a bivariate Gaussian the mutual information can be computed directly from 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 the correlation structure of a Gaussian random vector, we also know the mutual information between every pair of variables. We shall use this in the following made-up example. Namely, define a Gaussian chain in terms of 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.

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

It follows that the marginals have variance 1 as clearly $\mathrm{Var}[X_1] = \mathrm{Var}[Z_1] = 1$ and for $i > 1$, $\mathrm{Var}[X_i] = \vec{\rho}^2_{i-1,i}\mathrm{Var}[X_{i-1}] + \left(1 - \vec{\rho}^2_{i-1,i}\right)\mathrm{Var}[Z_i] = 1$ 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

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

This 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

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

which by induction implies $\rho_{i,j} = \prod_{k=i+1}^{j}\vec{\rho}_{k-1,k} = \rho_{j,i}$. 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. $f(x_i \mid X_{i-1}, X_{i-2}, \ldots, X_1) = f(x_i \mid X_{i-1})$. 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 correlations as elements of G_{obs}. In particular, the (i,j) entry of G_{obs} is $\rho_{i,j} = \prod_{k=i+1}^{j} \vec{\rho}_{k-1,k}$ when $i < j$ and 0 otherwise. This makes G_{obs} strictly upper triangular. 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. The reason for this becomes clear when we try to convolve G_{dir} based on the initial definition of a general Gaussian chain in Equation 1.1. We note that in **??** we usually (without an assumption of the topology of the random variables) use a symmetrical G_{obs}. We shall however postpone this discussion a bit and first use an upper triangular G_{obs}. In particular, we shall observe that we perfectly recover the *directional* correlations $\vec{\rho}_{k-1,k}$ from Equation 1.1 through **??**.

As G_{obs} is in this case strictly upper triangular, the spectral radius is 0 and hence we have no problems with convergence 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

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

Now, let G_{dir} be given as follows

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

then G_{dir}^2 is given by

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

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 random variables corresponding to the structural causal model, we completely recover (without any error) the

direct dependencies/correlation from the initial definition in Equation 1.1. This result holds for a general *chain* where Z_i can follow any distribution as long as they are uncorrelated. This follows from the above computation of $\text{Cov}[X_i, X_j]$, where no assumption of Z_j was needed except for correlation 0.

From the above, we might think that if we have a topological ordering of the random 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 the joint probability density function of two variables resembles a parabola. Namely, let $X_1 \sim \mathcal{U}(0,1)$ and $X_2 \mid X_1 \sim \mathcal{N}\left(1 - 4(x_1 - 1/2)^2, \sigma^2\right)$ i.e. $X_2 = 1 - 4(X_1 - 1/2)^2 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. 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 see a relationship between the two variables. 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 note, mutual information is a

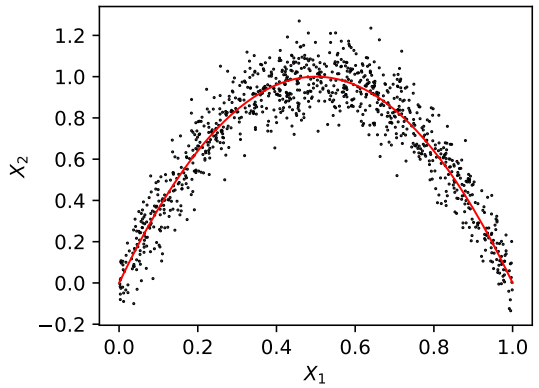


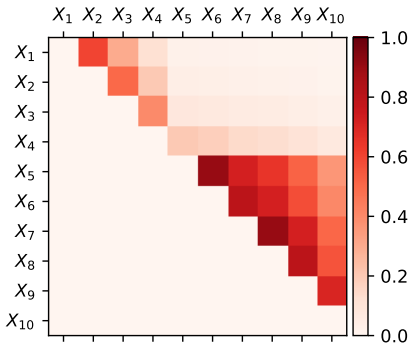
Figure 1.2: 1000 samples generated from $X_1 \sim \mathcal{U}(0,1)$ and $X_2 \mid X_1 \sim \mathcal{N}\left(1 - 4(x_1 - 1/2)^2, \sigma^2\right)$ with $\sigma = 1/10$. The mutual information is calculated theoretically to be $I(X_1, X_2) \approx 1.030$ and repeated simulations show that the empirical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

more general measure of correlation i.e. not just linear correlation. We proceed

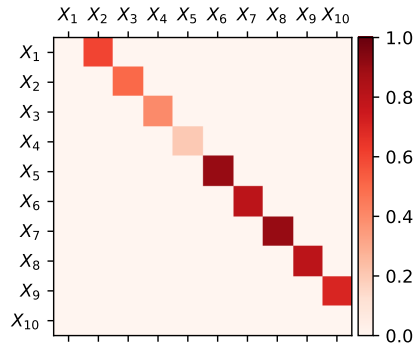
with a 10-Gaussian chain defined by the following correlations:

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

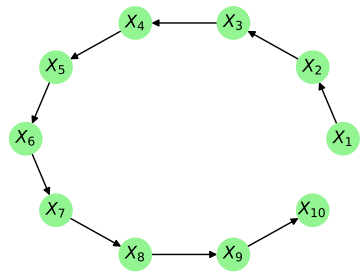
We have chosen correlations of different sizes to check if the deconvolution is robust in the presence of both strong and weak links. In particular, X_5 is only $\rho^2_{4,5} = 4\%$ of X_4 and the remaining 96% is noise/indescribable variance i.e. a very weak link between the first part of the chain 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.



(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.

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 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.

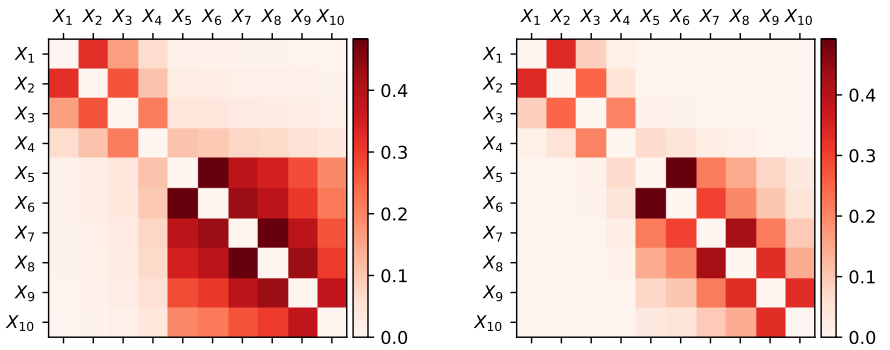
But first, we shall discuss the errors being made using the symmetric G_{obs} and G_{dir} instead of triangular. Namely, we investigate the powers of G_{dir}:

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

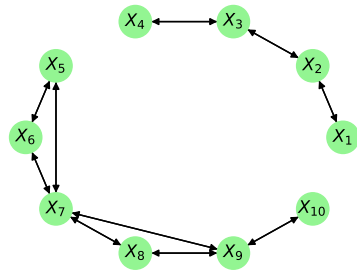
Higher power can be calculated similarly, but for the second power, we already observe an error. The first two terms correspond to a reflection of the second-order effects that we saw above and know to be true, whence the final two terms, that add to a diagonal matrix, is an error and will propagate 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.



(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.

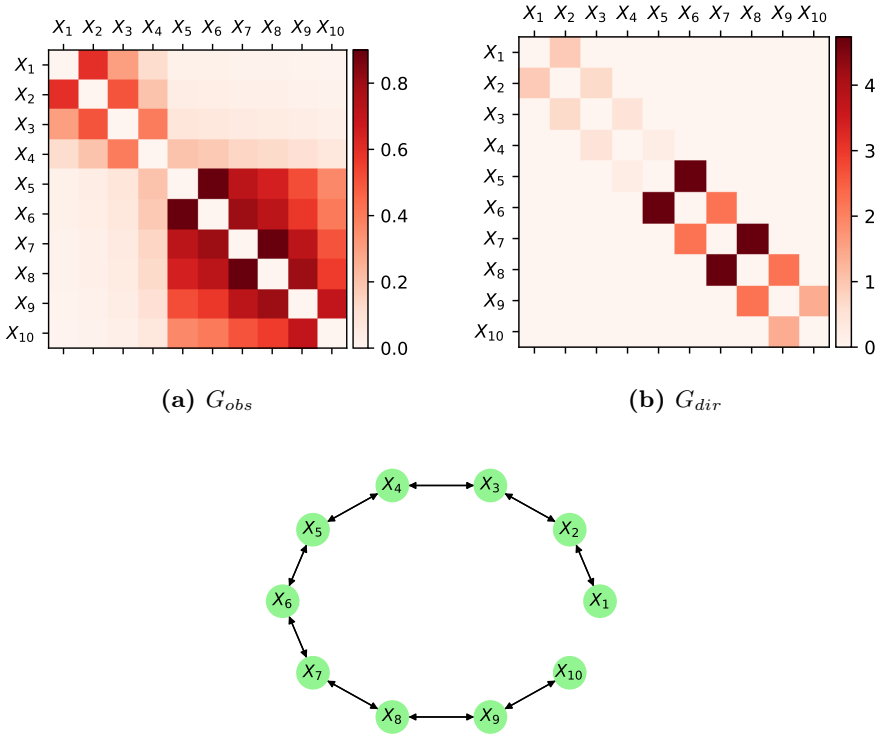
From Figure 1.4(b) we see that some correlation/association seem to bleed to variables 2 or 3 edges away which we of course know is not true given the Markov property discussed above. However, it is also clear that the error here is that the original assumption does not hold since using a symmetric G_{obs} implies that the measure of similarity flows both ways where in this case it is very much unidirectional.

We 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 true structure emerges as can be seen from Figure 1.5. Although the *correlations* in Figure 1.5(b) are not really correlations they do resemble those discovered in Figure 1.3(b). 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 without rescaling G_{obs} when discovering 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 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 can perfectly rediscover the network of direct dependencies. However, as noted above, correlation is not always a good measure of similarity. Thus, we proceed to experiment with mutual information on the same Gaussian chain.



(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.

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 $-1/2 \log\left(1 - \rho^2_{i,i+1}\right)$ and similarly, using Equation 1.2, we have that

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

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

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

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

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

In general, for a Gaussian chain, we have that

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

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 extent which we shall also see shortly when applying the deconvolution algorithm. Notice that as only the absolute value of the correlation matters, we only show the error for $\rho_{1,2}, \rho_{2,3} \geq 0$.

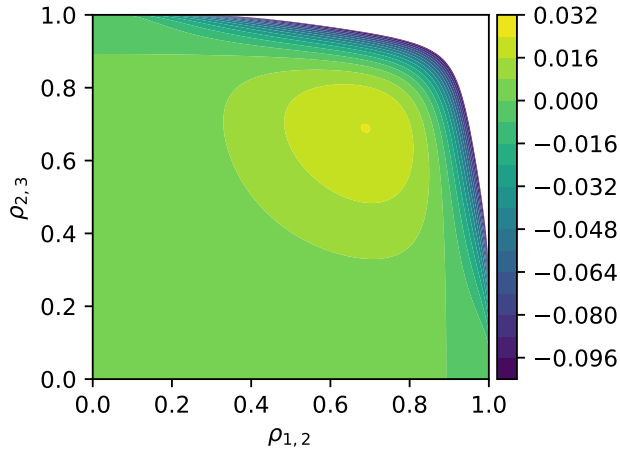
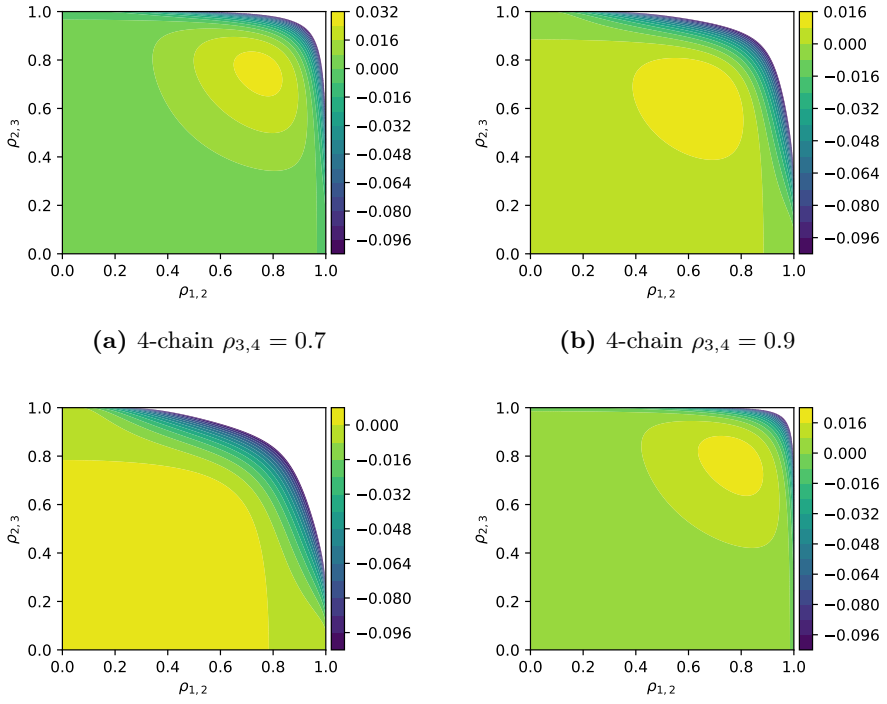


Figure 1.6: The error made by the assumption of G_{obs} and G_{dir} for second order observed effect. Although mutual information does not comply with the underlying assumptions, we observe that in the case of a Gaussian 2-chain, we can expect the error to be relatively small.

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 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 many-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 1.7(d), 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 affect 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 the 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.



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

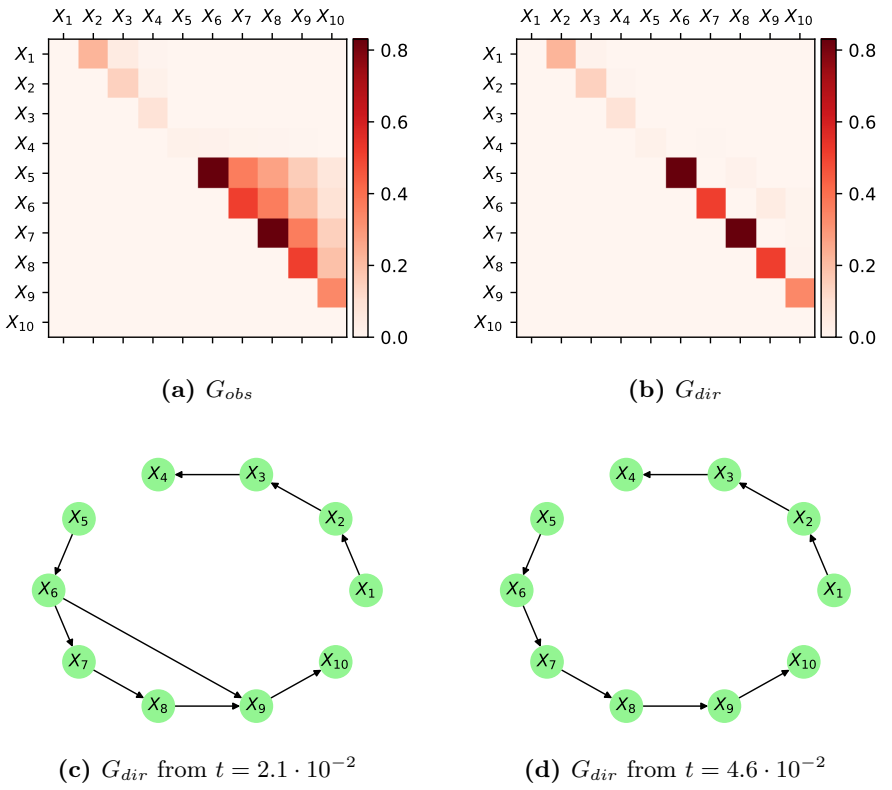
(b) 4-chain $\rho_{3,4} = 0.9$

(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.

However, we observe the same tendency to miss weak connections as was also observed in Figure 1.4. In total, we get excellent 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}

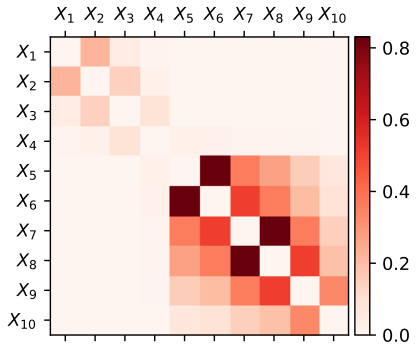
(c) G_{dir} from $t = 2.1 \cdot 10^{-2}$ **(d)** G_{dir} from $t = 4.6 \cdot 10^{-2}$

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.

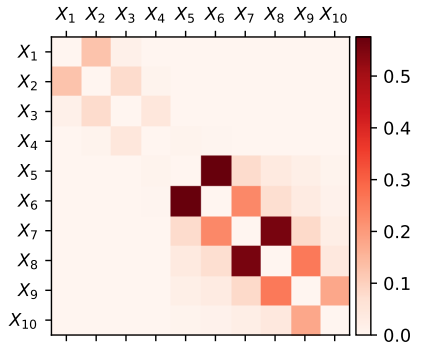
Finally, we use the corresponding symmetric G_{obs} (rescaled such that the largest absolute eigenvalue of G_{dir} is 0.99) which results in G_{dir} and the graph using a threshold $t = 4.88 \cdot 10^{-2}$ shown in Figure 1.9. Again, we observe some bleeding on the more strongly connected sub-chain as 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

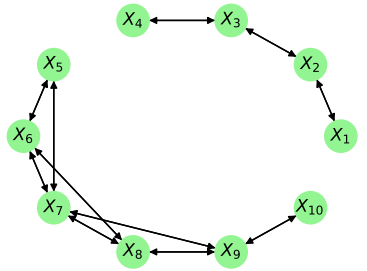
topological ordering of the variables, in which case these are heavily reduced as seen in Figure 1.3 and Figure 1.8. Thus, we proceed in the next section by considering a more complicated underlying (Gaussian) network to observe if other unwanted effects can occur and if a topological ordering is necessary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



(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}.

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. Such a DAG has a topological ordering. 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 from 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 labeling 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.

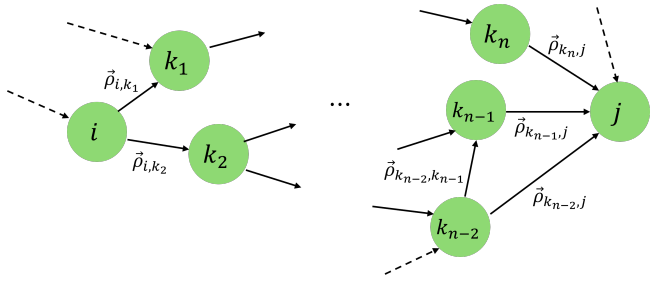


Figure 1.10: A general (linear) network represented as a DAG. The directed edge weights $\vec{\rho}_{k,l}$ specify how much the variable index k make up of variable l. Although i and j are not directly connected, multiple paths may exist between the two nodes, making the propagation of similarity more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

From the definition of the labels, it is clear that G_{dir} is once again strictly upper triangular as entries below the diagonal correspond 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 $\mathrm{Var}(X_i) = 1$. This is done to make the analysis later on simpler as then covariance and correlation are equal and $\vec{\rho}_{i,j}$ becomes 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 that 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 $k < i$ has 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 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 have 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.

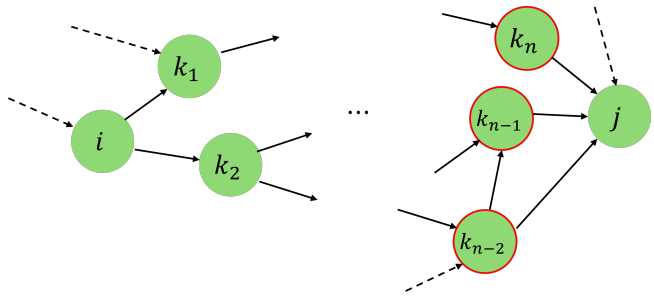


Figure 1.11: For node j, the set N_j^- is illustrated with red borders. It is exactly the set of nodes going directly into j. We note that an in-neighbor l of in-neighbor k of node j can also be an in-neighbor of j i.e. l can influence both k and j whilst k also directly influenced j. It is in particular 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. 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 can 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.

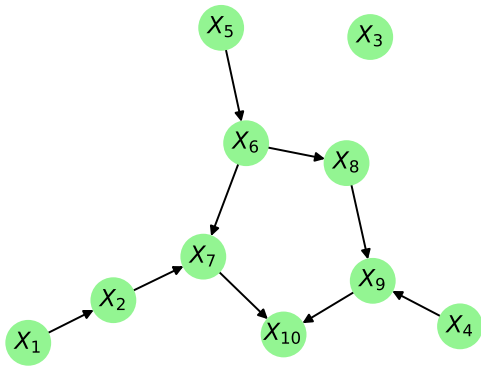
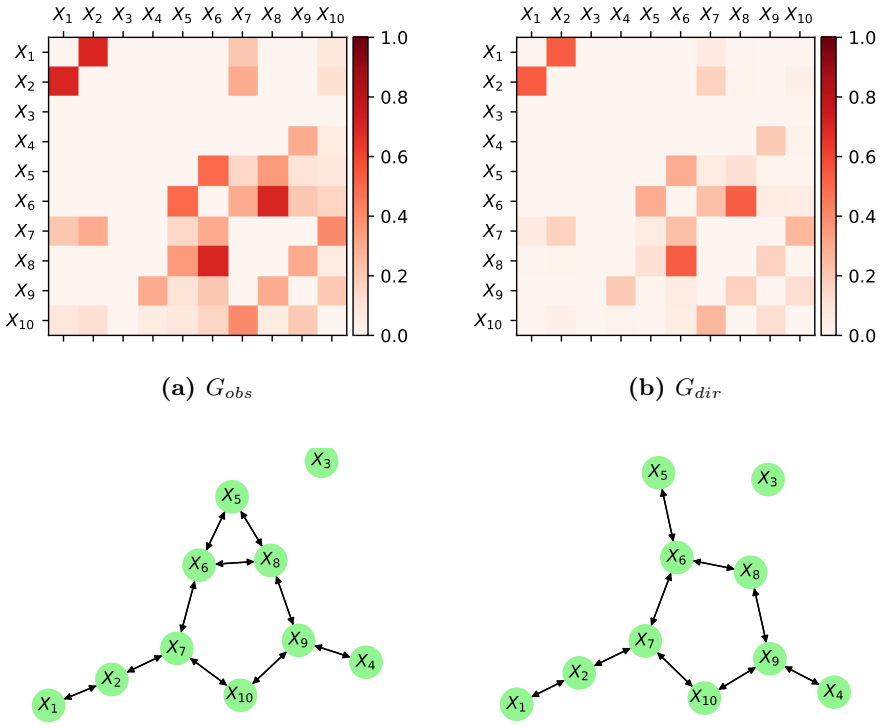


Figure 1.12: The graph defined in Equation 1.7. Note that X_3 is neither influenced nor influences any other variable. It is of course in our interest to accurately tell if X_3 should be considered if we try to infer a probability distribution on e.g. X_{10} given observations of the other variables.

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



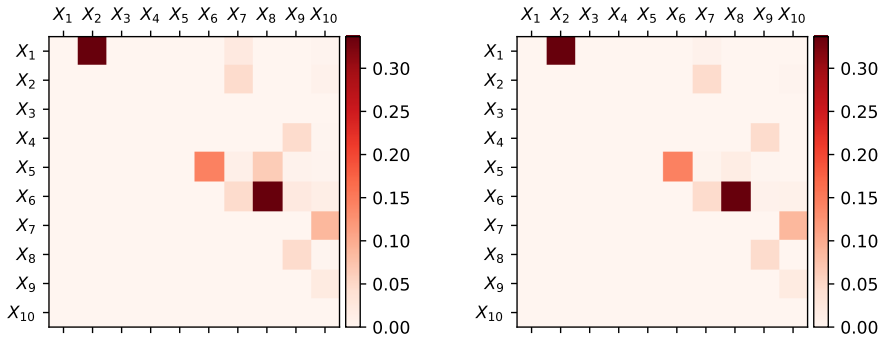
(a) G_{obs} **(b)** G_{dir}

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

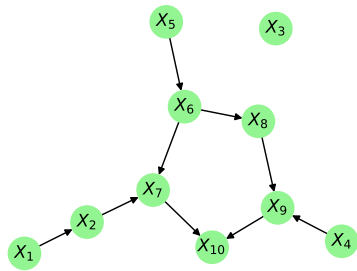
Figure 1.13: Not knowing the topological structure and thus using a symmetric G_{obs} (a) we obtain the G_{dir} in (b). Clearly, there is some bleeding, but choosing the threshold $t = 1.18 \cdot 10^{-1}$ we can accurately rediscover the network structure up to a direction on the edges. As with the previous 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.

Finally, before 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 compared 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 1.14(c)) 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 both of them.



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

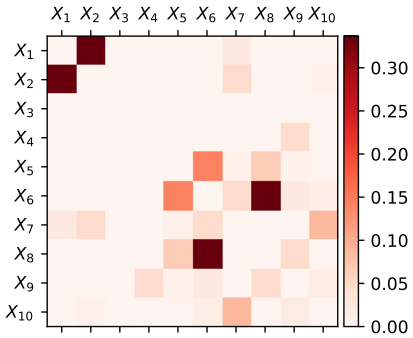


(c) G_{dir} as a graph

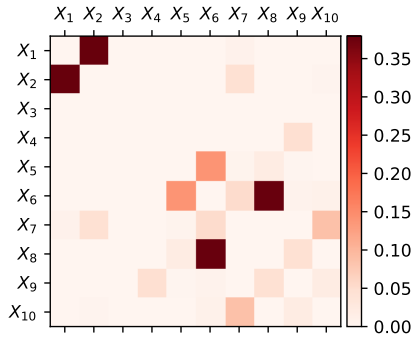
Figure 1.14: Using mutual information instead of correlation results in G_{obs} shown in (a). The non-linear map from correlation to mutual information only effects the resulting G_{dir} a little as shown in (b) when comparing to the $\vec{\rho}_{i,j}$ from Equation 1.7. Choosing the relatively small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

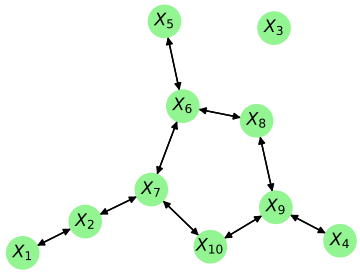
does not seem to have much of an effect as seen from Figure 1.15. Although there still is a small weight on the edge from X_5 to X_8, by choosing the threshold $t = 1.96 \cdot 10^{-2}$ we can accurately construct the true network structure.



(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 $t = 1.96 \cdot 10^{-2}$.

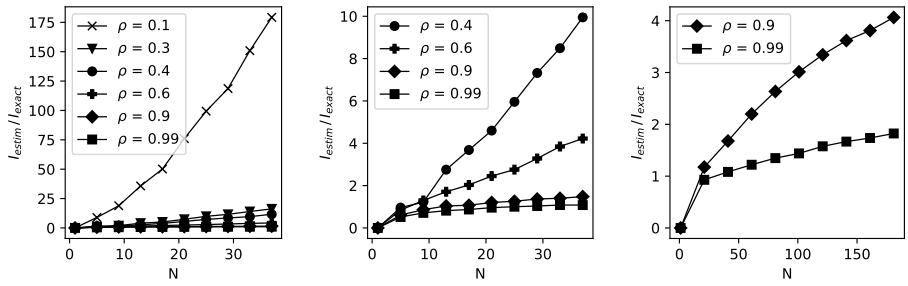
In conclusion, we observe a useful property of more general networks that for both mutual information and correlation, the additional assumption of the topological order does not have much of an effect in these cases contrary to what we observed for Gaussian chains and linear chain models in general,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

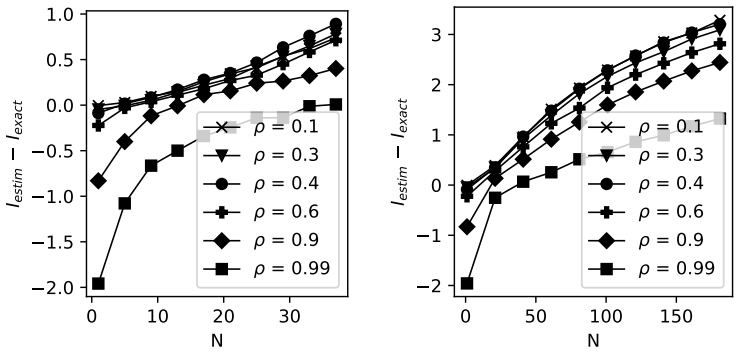
1.3.1 Spline and KDE based CE estimation

Before the first example, we shall 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 use a raw binning approach, the number of bins N influences the estimate a lot, and no number of bins seems to perform well in all cases. Namely, let $\boldsymbol{X}$ be a bivariate Gaussian with correlation ρ, then the Copula density looks as in Figure 1.21(c). In particular, we notice the peaks at $(0, 0)$ and $(1, 1)$ from which most of the mutual information originates. 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. $N = 25$ corrects this error for large correlations, while small mutual information for small correlations remains relatively small. However, when deconvolving we would preferably want a more precise way of estimating the mutual information.



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

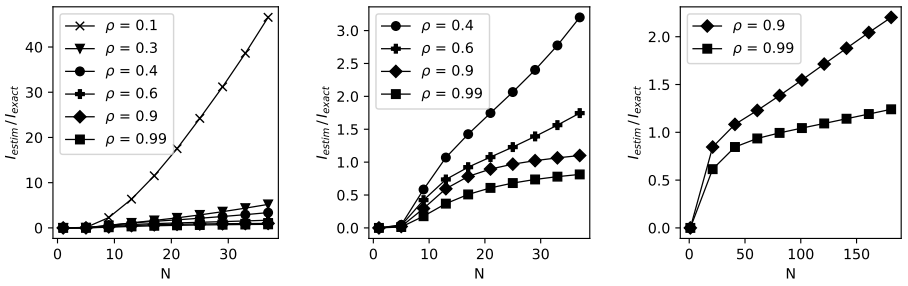
Figure 1.16: Relative error when estimating mutual information for different bivariate Gaussian distributions with varying bin counts N. Information originating from small correlations are vastly overestimated.



(a) **(b)**

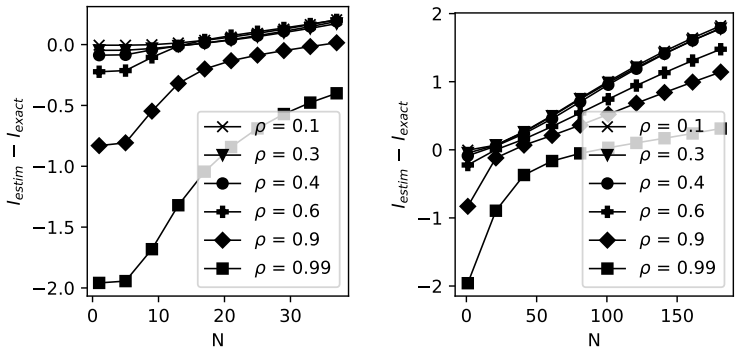
Figure 1.17: Error for mutual information estimation for varying correlations. Contrary to the relative error, we see that it is the mutual information from large corrlations that is the most error-prone for small bin counts.

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 the 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 for large N but also that a better choice for the number of bins is around $N = 50$ contrary to the results of [**?**].



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

Figure 1.18: Relative error of mutual information estimates using B-splines. Comparing to the relative error of the raw histogram based approach, we obtain relative error much smaller.



(a) **(b)**

Figure 1.19: The actual error for mutual information estimation using the B-splines approach. For $n = 400$ observations, performance is comparable to that of the raw histogram based approach although a larger bin count is preferred.

The results for M-splines are shown in **??** 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 information is large, most of the mutual information comes from small domains at $(0, 0)$ and $(1, 1)$. Hence, to calculate the mutual information to a high precision, we 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 were obtained through the theoretical Copula density function evaluated at the bin centers $\left(\frac{2i-1}{2N}, \frac{2j-1}{2N}\right)$ for $i, j \in \{1, \dots, N\}$. 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 $n = 400$ observations to evaluate the performance of the KDE from the previous chapter. We note that a more memory-efficient implementation is possible by splitting up the computation into multiple parts as the problem with many observations and bins is that the computation is $\mathcal{O}\left(N^2 n\right)$ time and memory if done all at once.

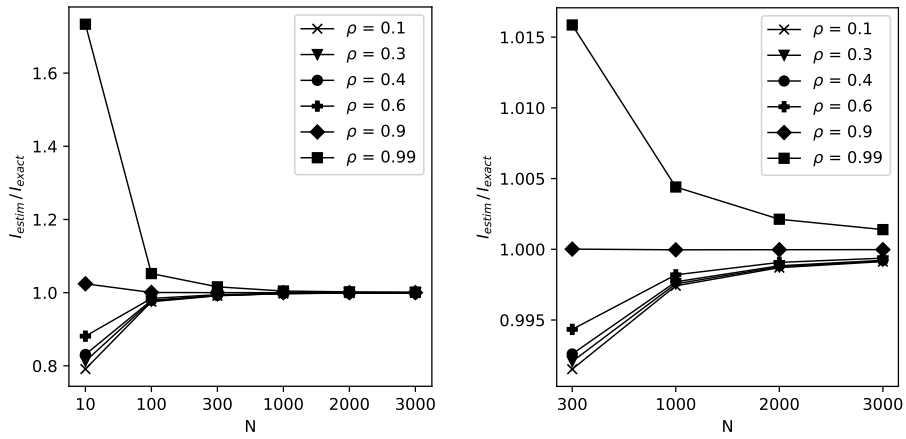


Figure 1.20: Relative error of mutual information based on the true Copula density. We observe that for $N \geq 300$ the relative error is almost negligible.

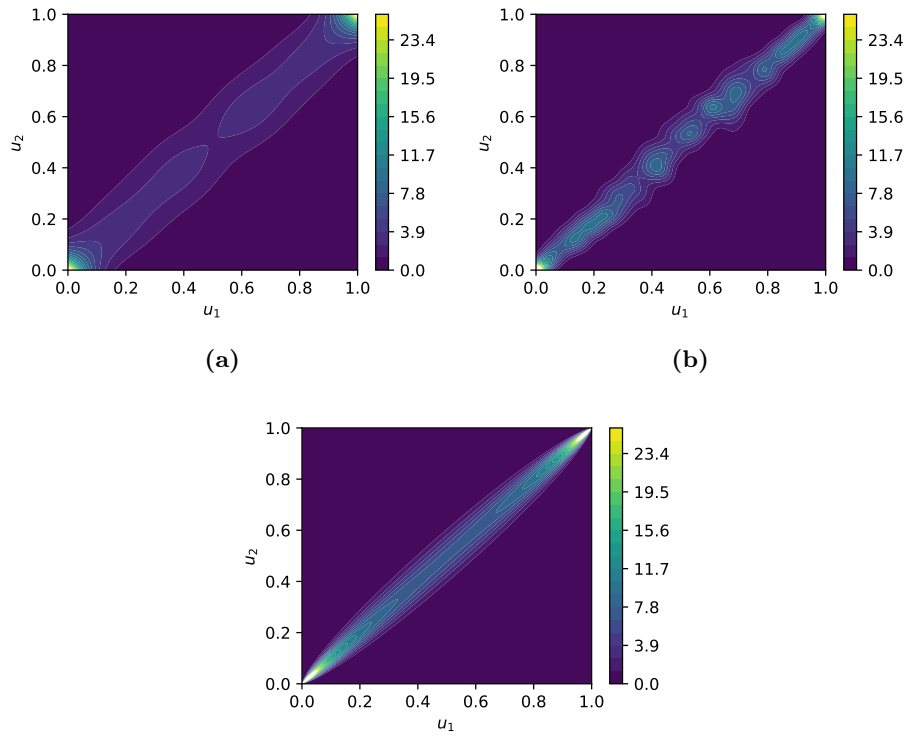
In Table 1.1, we have used the boundary-corrected KDE from **??** to first estimate the Copula density function and then estimate the mutual information from this. We note that by default, the bandwidth is chosen to be $h = h^{Scott} = 0.085$ as the marginals are approximately uniform and hence the variance is constant.

From Table 1.1, we observe relatively low variance, and in general, we compute the mutual information to a higher accuracy than both 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

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: Estimating mutual information based on $n = 400$ samples from different bivariate Gaussians using the boundary corrected KDE. Repeating the simulations 10 times, we obtain average errors and variance of the estimate. In particular, the estimates are very certain for $n = 400$. However, as for the spline based methods, large correlations result in underestimating the mutual information. This can however be corrected by tuning the bandwidth as is also observed in the above.

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 as the 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 the estimate, observed 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 (as discussed in **??**) 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 close to perfectly correlated, the KDE-based method performs quite well. This, we shall also see in the following section where we couple the above discussions on mutual information estimation with the deconvolution algorithm.



(a) **(b)**

(c)

Figure 1.21: Estimated Copula densities (a) and (b) compared to the theoretical Copula density (c) for $\rho = 0.99$. Using a smaller bandwidth we are able to capture the peaks at $(0, 0)$ and $(1, 1)$ more accurately but at the cost of undersmoothing the interior.

1.3.2 Exponentiated multivariate Gaussian

In this section, we will consider a small example, testing the combined algorithm. In particular, we shall discover what to be aware of when data is not easily transformed by the inverse distribution function. Let us consider a simple case with $\mathbf{Y} = e^{\mathbf{X}}$ (element-wise exponentiation) where $X \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0.9\sigma_1\sigma_2 & 0 \\ 0.9\sigma_1\sigma_2 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix} = \operatorname{diag}(\boldsymbol{\sigma}) \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \operatorname{diag}(\boldsymbol{\sigma}) \tag{1.8}$$

In particular, in terms of Equation 1.4, we have that for $\boldsymbol{X}$, $\vec{\rho}_{1,2} = 0.9$. From **??**, it is clear that the (symmetric) mutual information matrix G_{obs} is the same as of $boldsymbol{X}$ and hence by Proposition 1.1 is as follows

$$G_{obs} = \begin{bmatrix} 0 & -\frac{1}{2}\log\left(1-\vec{\rho}_{1,2}^2\right) & 0 \\ \log\left(1-\vec{\rho}_{1,2}^2\right) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \approx \begin{bmatrix} 0 & 0.83037 & 0 \\ 0.83037 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In particular, independent of the choice of $\boldsymbol{\sigma}$, we should observe an estimated $\hat{G}_{obs}$ close to this. We choose the following three cases for the choice of $\boldsymbol{\sigma}$.

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

To draw from this distribution, one can use built-in functions or the Cholesky factorization of the correlation matrix to generate 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. Once again, we shall only use 400 samples as this resembles the number of observations in the pharmaceutical dataset, which we shall treat in Section 1.4. We note that a KDE has been used to approximate the distribution function. We will see shortly why this is not always a good idea.

For $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$, **??** returns the following

$$\hat{G}_{obs} = \begin{bmatrix} 0 & 0.8618 & 0.07889 \\ 0.8618 & 0 & 0.07880 \\ 0.07889 & 0.07880 & 0 \end{bmatrix} \tag{1.9}$$

Similarly, for $\boldsymbol{\sigma} = (1, 1, 1)$:

$$\hat{G}_{obs} = \begin{bmatrix} 0 & 1.066 & 0.1667 \\ 1.066 & 0 & 0.1825 \\ 0.1667 & 0.1825 & 0 \end{bmatrix} \tag{1.10}$$

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

$$\hat{G}_{obs} = \begin{bmatrix} 0 & 1.797 & 1.549 \\ 1.797 & 0 & 2.145 \\ 1.5492 & 2.145 & 0 \end{bmatrix} \tag{1.11}$$

Note that for this example, we have chosen $h = \frac{1}{3}h^{Scott}$ to correct for the behavior observed in Table 1.1 when computing the mutual information between X_1 and X_2.

For $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ we observe the most resemblance to the theoretical G_{obs}. For the latter two examples, $\boldsymbol{\sigma} = (1, 1, 1)$ and $\boldsymbol{\sigma} = (1, 2, 3)$, we see a completely different result and immediately suspect that there must be some errors either in the algorithm or the underlying assumptions of the algorithm. Investigating the partial results of **??** we immediately see a flaw in the supposedly uniform variables U_i as shown in figure Figure 1.22 for $\boldsymbol{\sigma} = (1, 2, 3)$, which through a Kolmogorov Smirnov test are indeed significant (Table 1.2).

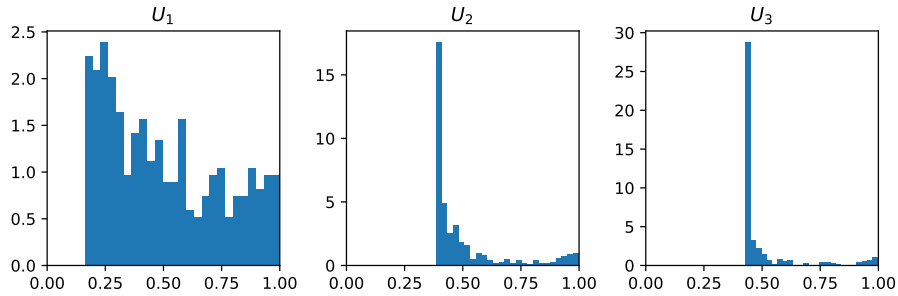


Figure 1.22: The samples transformed using $U_i = F_i(Y_i)$ for $\boldsymbol{\sigma} = (1, 2, 3)$. These should be uniformly distributed, but clearly this is not the case for neither of them, as we have demonstrated in Table 1.2.

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

Table 1.2: Kolmogorov Smirnov test result based on 400 samples for $\boldsymbol{\sigma} = (1, 2, 3)$. It is clear that all samples are statistically significant.

Before handling this, the non-uniformity of U_1, U_2 and U_3 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 which is further shown statistically significant in Table 1.3.

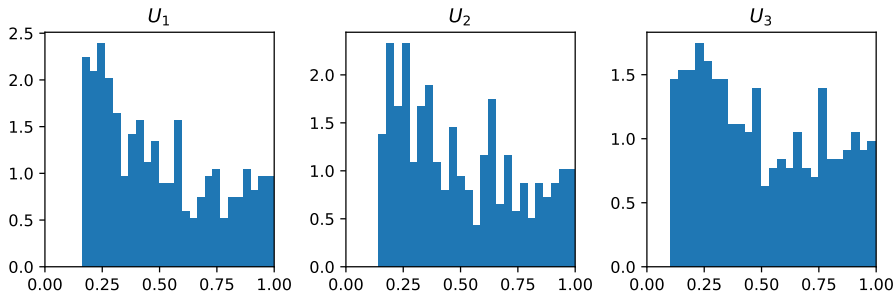


Figure 1.23: The samples transformed using $U_i = F_i(Y_i)$ for $\boldsymbol{\sigma} = (1, 1, 1)$.

	U_1	U_2	U_3
D_n	0.16511	0.14672	0.10561
p-value	0	0	$2.382 \cdot 10^{-4}$

Table 1.3: Kolmogorov Smirnov test result based on 400 samples for $\boldsymbol{\sigma} = (1, 1, 1)$. Once again, all the transformed samples are shown statistically significant.

Finally, for the sake of completeness, $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ is also shown in Figure 1.24 and seems very reasonable, except for U_3 which again, is shown through the Kolmogorov Smirnov test in Table 1.4 to be significant.

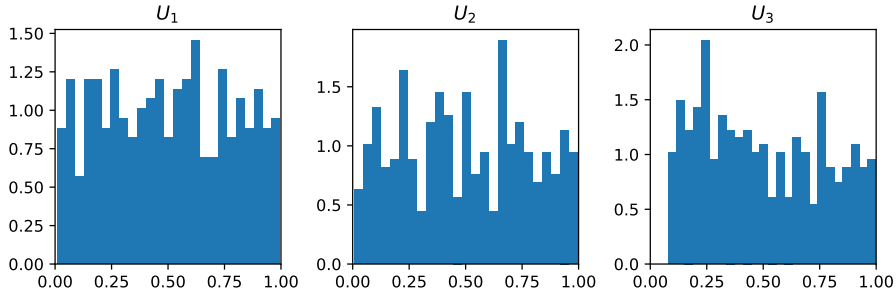


Figure 1.24: The samples transformed using $U_i = F_i(Y_i)$ for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

From the above examples, it seems that the larger the variance, the worse the uniforms turn out. From Figure 1.25, we see that this is primarily due to a poor fit of the KDE, where we have zoomed in on the interval $[-200, 200]$ which contains 96.2% of observations. In particular, the peak neat $y_3 = 0$ is not

	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: Kolmogorov Smirnov test result based on 400 samples for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

captured by the KDE. The poor fit is primarily due to the use of Scott's Rule which in this case overshoots the optimal bandwidth by a lot. 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 for the KDE lies above 0.

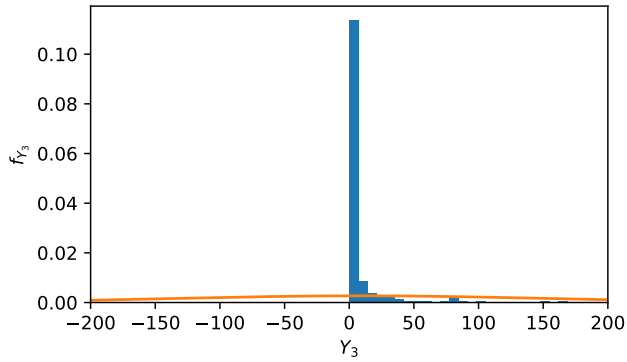


Figure 1.25: Inspecting the fit of the KDE on Y_3, we indeed observe that it is quite poor. The peak near 0 is not captured. Thus, either Y_3 has to be transformed through some strictly positive function or another estimate of the distribution function (such as the empirical distribution function) need to be applied.

By **??**, we can get rid of these numerical issues by transforming Y_i using e.g. $\log(\cdot)$ or $(\cdot)^p$ for $p > 0$ to even out the observations more. As the first simply inverts the initial transformation of $\boldsymbol{X}$, 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 $\boldsymbol{Y}$, resulting in $\boldsymbol{Y}^p$, prior to running **??**. The resulting samples $u_i^{(j)}$ are shown in Figure 1.26 along with statistical test for uniform distribution in Table 1.5

The resulting $u_i^{(j)}$ are now no longer significant and indeed the KDE fits much better as seen in Figure 1.27. Furthermore, the estimated $\hat{G}_{obs}$ in Equation 1.12 is similar to the one obtained from $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ with the notable differ-

ence $I(Y_1, Y_3)$ and $I(Y_2, Y_3)$ are closer to the true value, 0.

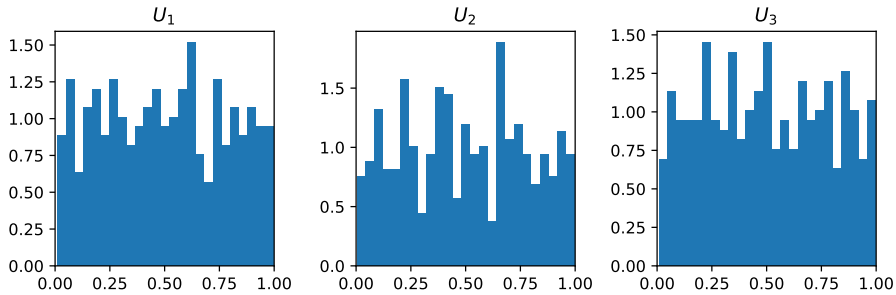


Figure 1.26: The resulting observations of $\boldsymbol{U}$ after a power transformation with $p = 1/10$ has been applied to $\boldsymbol{Y}$.

	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 400 samples of $\boldsymbol{Y}^p$ with $\boldsymbol{\sigma} = (1, 2, 3)$ neither are statistically significant.

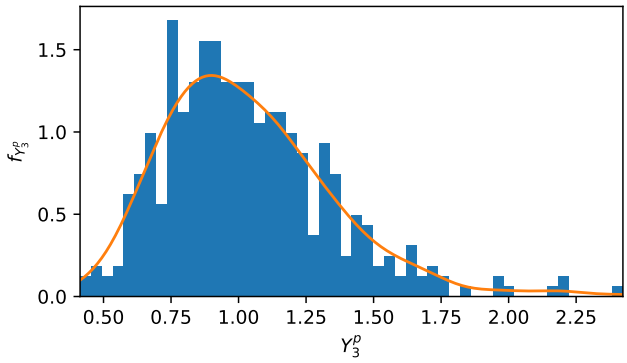
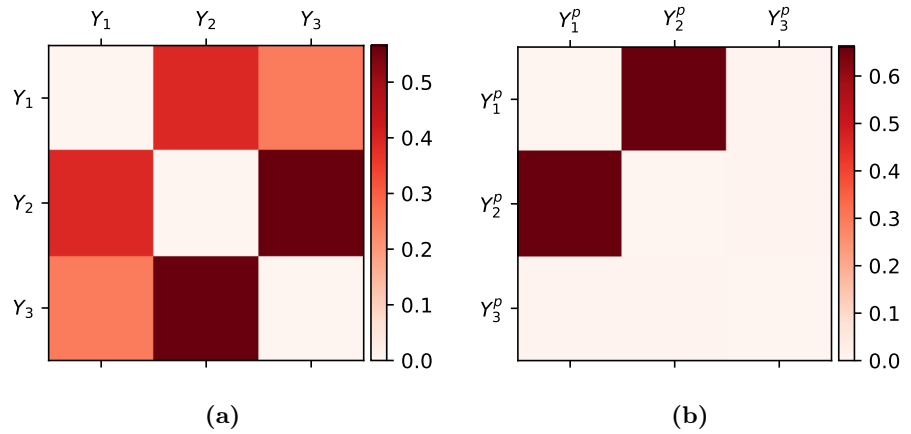


Figure 1.27: The KDE fit on Y_3^p with $p = 1/10$. Now, the KDE fits the samples much better, leading to non-significant results regarding tests of uniformity.

$$\hat{G}_{obs} = \begin{bmatrix} 0 & 0.8629 & 0.01799 \\ 0.8629 & 0 & 0.01886 \\ 0.01799 & 0.01886 & 0 \end{bmatrix} \tag{1.12}$$

To illustrate the importance of the above discussion, although it should already be quite clear from the differences in $\hat{G}_{obs}$, we have in Figure 1.28, shown the resulting $\hat{G}_{dir}$ after using **??**. Namely, we infer a relation between Y_2 and Y_3 especially, which we know to be independent of each other from Equation 1.8.



(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 $\boldsymbol{Y}^p$.

We note that from **??**, we can compute a CI of the absolute value of the correlation (under the assumption the mutual information was computed from a bivariate Gaussian or a strictly increasing transform of such variables) based on [**?**]. In particular, computing confidence intervals for the absolute correlation from each mutual information from $\hat{G}_{0bs}$ in Equation 1.12, we see that the estimated $\widehat{I(Y_1, Y_2)}$ is not significantly different from the theoretical mutual information while the remaining are (i.e. between (Y_1, Y_3) and (Y_2, Y_3) respectively).

Thus, in the above, we have shown that the assumption of uniform marginals does not always hold when the data has heavy tails. Namely, the key assumption for **??** is that we are doing computations on a Copula density such that the entropy of the marginals $h(Y_i)$ and $h(Y_j)$ is 0. Thus, when this assumption does not hold, the algorithm does estimate the mutual information correctly. However, we can fix this by adding the marginal entropies to the algorithm using **??**. However, for the KDE to estimate the marginal entropies well, we still would not want a tail too heavy. What a too-heavy tail is, is a bit ambiguous, but using the modified algorithm on $\boldsymbol{\sigma} = (1, 2, 3)$ without the power transformation we obtain the following $\hat{G}_{obs}$, which is of course not as good as the above $\hat{G}_{obs}$,

but still closer than the original estimate, in Equation 1.11

$$\hat{G}_{obs} = \begin{bmatrix} 0 & 0.6263 & 0.02288 \\ 0.6263 & 0 & 0.01688 \\ 0.02288 & 0.01688 & 0 \end{bmatrix}$$

In the following section, we shall apply our learning from this section to the more complicated yet fully controlled example, introduced in Section 1.2. In particular, we shall observe that without any manual tuning of the bandwidths h, we are able to accurately infer the causal structure.

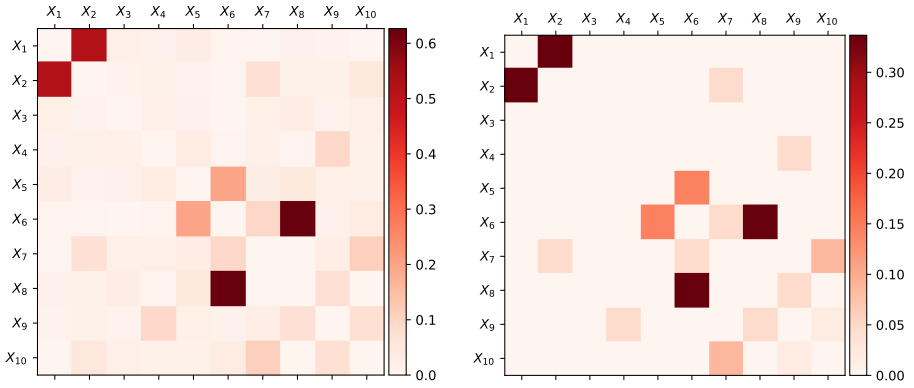
1.3.3 Gaussian network revisited - Application of complete framework

In this section, we will redo the example from Section 1.2 where the underlying causal structure was defined in Equation 1.7. However, this time, we shall first simulate $n = 400$ observations from the joint distribution and then estimate the Copula density based on these observations instead of using the theoretical Copula densities (i.e. the exact mutual information) along with the learnings from the previous sections. In particular, we shall observe that the algorithm for estimating the mutual information between pairs of random variables performs well enough for us to recover the structure perfectly. As the largest (direct) correlation is 0.7, we expect from Table 1.1 that the mutual information will be accurately estimated.

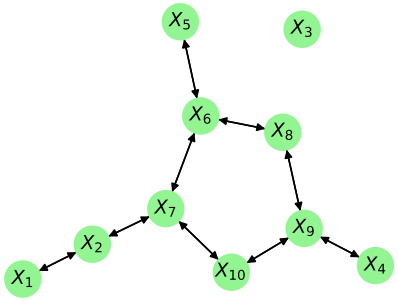
In Figure 1.29, we have summarized the results using a symmetric G_{obs}. In particular, we observe that the estimated $\hat{G}_{dir}$ is nearly identical to the true G_{dir} (from Equation 1.7). Furthermore, choosing the relatively small threshold $t = 0.06$ we recover the true causal structure although the entries of $\hat{G}_{dir}$ appear to be a factor 2 as large as the true G_{dir}. If we instead assume a topological structure, resulting in a triangular $\hat{G}_{obs}$, this deviation is alleviated as seen in Figure 1.30.

From this short example of combining **??** and **??**, we observe that indeed the methodology proposed can at least for networks as defined in Section 1.2 recover the causal structure. Furthermore, from **??**, we note that the same structure would be inferred if instead the variables had been transformed by a strictly increasing function. This is particularly important in the following section where we shall use the framework on the data introduced in **??**.

As an example, suppose that a network is defined as in Section 1.2, but we only observe $\boldsymbol{Y}$ where $Y_i = f(X_i)$ with f being strictly positive. Then, the mutual information between Y_i and Y_j is the same as between X_i and X_j which we from this example is accurately deconvolved.

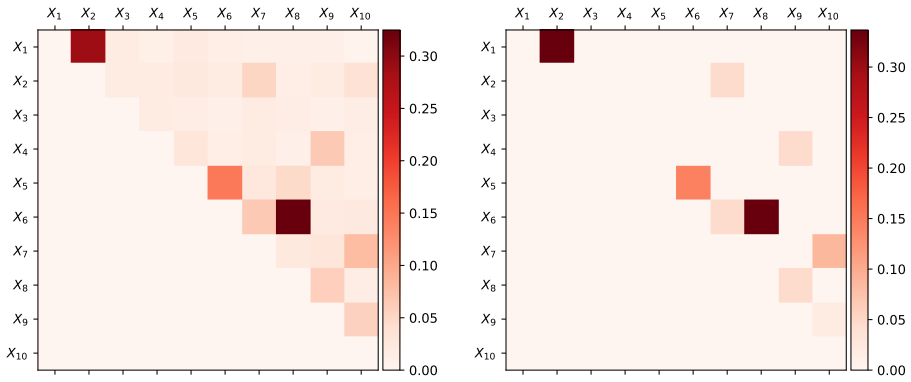


(a) $\hat{G}_{dir}$ **(b)** G_{dir}

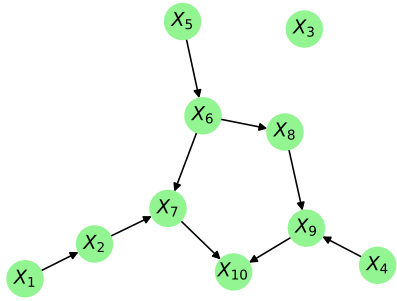


(c) $\hat{G}_{dir}$ as a graph

Figure 1.29: Combining the method for estimating the mutual information and algorithm for deconvolving the network (using a symmetric G_{obs}) we observe near-optimal results. In particular, the structure of $\hat{G}_{dir}$ (a) is very alike to $\hat{G}_{dir}$ (b). This is also shown in (c), where the $\hat{G}_{dir}$ is represented as a graph using $t = 0.06$, and we recover the original graphical structure. We observe that the entries of $\hat{G}_{dir}$ are almost twice the size of G_{dir}.



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



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

Figure 1.30: When assuming a topological order of the random variables, we observe that the inferred $\hat{G}_{dir}$ (a) is much more alike the true G_{dir} (b). Furthermore, using a threshold $t = 0.05$ we recover the true causal structure, removing the noise entries in $\hat{G}_{dir}$.

1.4 Pharmaceutical data deconvolution

Finally, we turn our attention to the pharmaceutical production data introduced in **??**. Namely, we have at this point used the methodology discussed in chapter 1 both to estimate mutual information, to deconvolve Gaussian networks and chains based on analytical expressions for the correlation between variables, and finally in Subsection 1.3.3 where we have combined **??** and **??** to infer the causal structure of a 10-dimensional Gaussian network.

In particular, we saw that the methodology combined produced very accurate results when the mutual information between pairs of random variables was not too large as these proved difficult to estimate to high accuracy without turning to a manual tuning of the bandwidths. We especially want to avoid the manual tuning of the bandwidth in this section, as we shall compute mutual information between 1653 pairs of random variables.

The random variables are the duration and delays of each process as well as the change in the level of the tank during these operations. Recall that e.g. $T_{4.1}^P$ is the duration of process 4.1 while T_4^D is the delay after all subprocesses of process 4 are completed. Also, for each *temporal* variable T_i^P and T_i^D, we have a corresponding change of level M_i^P and M_i^D. Initially, we also add the accumulated random variables. Namely, for process 1, we add the random variable $T_1 = T_1^P + T_1^D$ and likewise for the level changes for all processes. Also, the total duration $T = \sum_{i=1}^{10} T_i$ and change in level $M = \sum_{i=1}^{10} M_i$ are added as random variables. The accumulated random variables are initially added to see if our method can *rediscover* these simple causal relations. We will later remove these from our discussion as we try to gain a deeper understanding of the production system.

Hence, from the above, we initially have 68 variables. However, some of them are constant and thus out of interest. The constant *random* variables are

$$M_1^D, \quad M_2^D, \quad T_{4.1}^P, \quad M_{4.1}^P, \quad T_{4.3}^P, \quad M_4^D, \quad M_6^D, \quad T_8^P, \quad M_{10}^D$$

However, as $T_8 = T_8^P + T_8^D$ we shall also exclude either T_8 or T_8^D due to T_8^P being constant. In particular, T_8 and T_8^D can be interchanged at will when computing mutual information according to **??**. Here, we have chosen to keep T_8^D as it is easier to relate to the other processes. Thus, we are only considering 58 random variables and hence $58 \cdot 57/2 = 1653$ pairs of variables as stated above.

We note that as the resolution in time is 0.001 we add uniform distributed noise from $[0, 0.001]$ to the durations and delays. In particular, we observe that in some cases equal durations are observed. These would result in non-uniform

marginals like in Subsection 1.3.2 hence making the algorithm fail at computing the mutual information accurately. We note that the experiments to follow have been carried out multiple times to assess the influence of this perturbation on the durations and delays. However, as this did not influence our results in the slightest we will not discuss this further. Furthermore, instead of using a KDE as in Subsection 1.3.2 to transform the (continuous) observations to lie on the unit interval through the distribution function, we have used the empirical distribution function. This was done to ensure that all 58 random variables were transformed appropriately without having to consider transformations (like in Subsection 1.3.2).

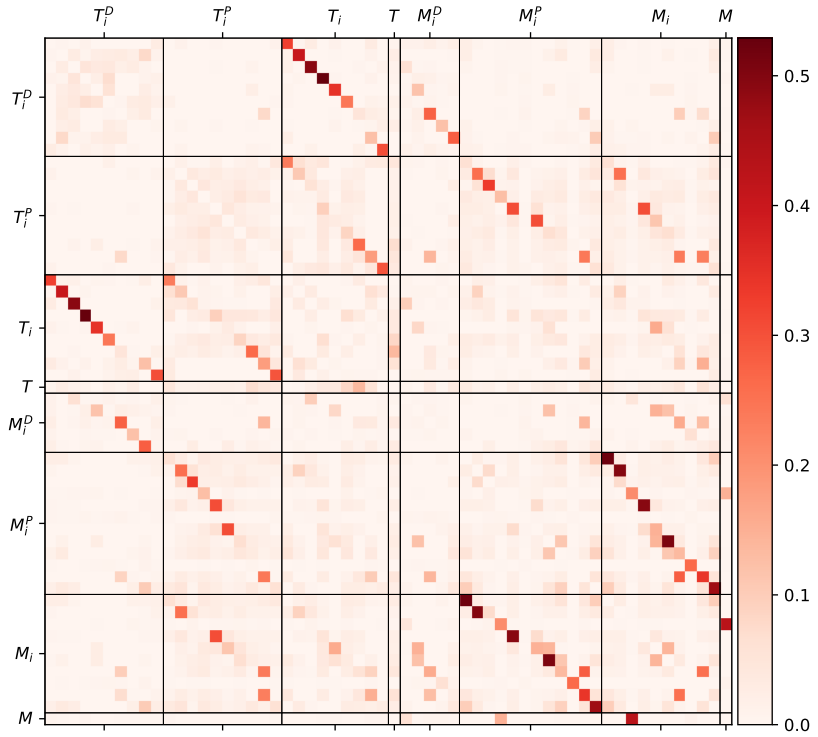


Figure 1.31: G_{dir} from a symmetric G_{obs}. Other than the accumulated variables, we observe strong dependencies between the duration of a process and the change in level during the process by comparing the columns and rows labeled T_i^P and M_i^P.

The resulting G_{obs} is shown in the appendix, **??**. We observe that there indeed is a strong dependence between the accumulated variables and their summands

such as T_1 and T_1^D as well as T_1^P. We immediately use the deconvolution algorithm resulting in the G_{dir} seen in Figure 1.31

We have labeled sections of G_{dir} such that each section corresponds to an ordered list of variables. The variables in each section are

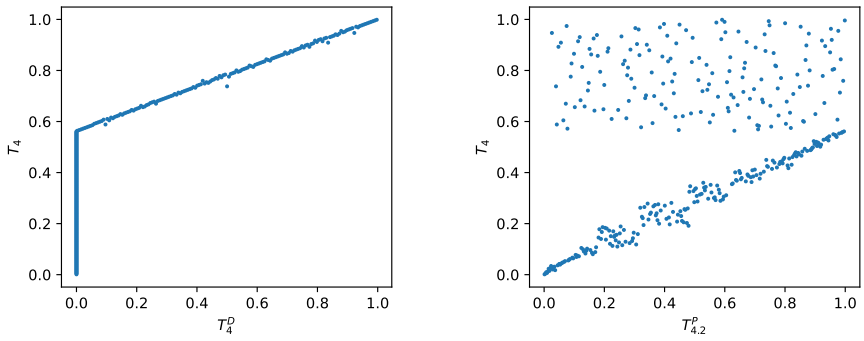
$$\begin{aligned}
T_i^D &= \left\{T_1^D, T_2^D, T_3^D, T_4^D, T_5^D, T_6^D, T_7^D, T_8^D, T_9^D, T_{10}^D\right\} \\
T_i^P &= \left\{T_1^P, T_2^P, T_{3.1}^P, T_{3.2}^P, T_{4.2}^P, T_5^P, T_6^P, T_7^P, T_9^P, T_{10}^P\right\} \\
T_i &= \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_9, T_{10}\} \\
T &= \{T\} \\
M_i^D &= \left\{M_3^D, M_5^D, M_7^D, M_8^D, M_9^D\right\} \\
M_i^P &= \left\{M_1^P, M_2^P, M_{3.1}^P, M_{3.2}^P, M_{4.2}^P, M_{4.3}^P, M_5^P, M_6^P, M_7^P, M_8^P, M_9^P, M_{10}^P\right\} \\
M_i &= \{M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}\} \\
M &= \{M\}
\end{aligned} \tag{1.13}$$

I.e. the section labeled T_i^D consists of all the delays after each process and so on for the remaining section labels.

From Figure 1.31, we observe that the accumulated durations of each process T_i are indeed very dependent on both the delay and duration of the process. This is a sign that the algorithm performs well as by definition $T_i = T_i^D + \sum_{k \in \mathcal{P}_i} T_i^P$ (where $\mathcal{P}_i$ is defined as the set of subprocesses that constitute the process i).

For example, we observe that T_4^D is strongly associated with T_4 while $T_{4.2}^P$ is not as strongly associated with T_4. However, there is still a connection. If we plot T_4 vs. T_4^D and $T_{4.2}^P$ respectively (Figure 1.32) we indeed observe that the information between T_4^D and T_4 is much greater than $T_{4.2}^P$ and T_4. As we saw in **??**, this is because the delays after process 4 are much larger than the duration to begin with. Thus, even though T_4^D is 0, 56.25% of times, most of T_4 can be explained from this alone when compared to the scores between T_4 and the other random variables.

Furthermore, we observe that the total duration T is mostly explained by T_7 which in turn is explained mostly by T_7^P. Although there are many other interesting observations such as the most change in levels and masses occurs during process 3, addition of a material, and stirring. The level after process 3 is in turn mostly influenced by the stirring but also seems to have some unexplained variation. We observe the unexplained variation from the row corresponding to M_3 which has relatively small associations with the other delays, durations, and levels.



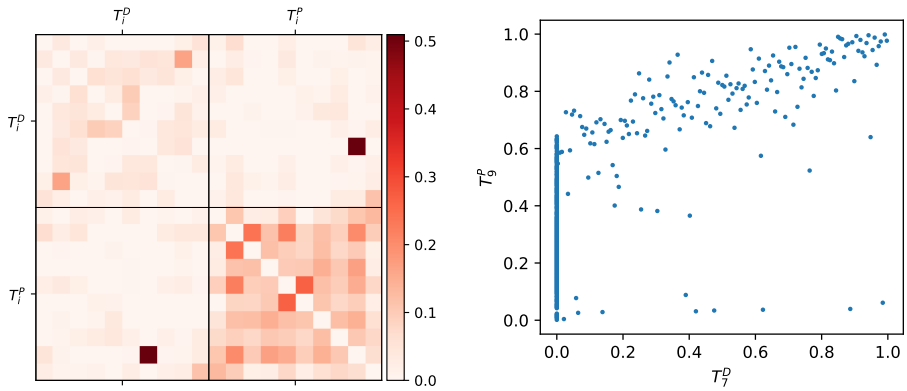
(a) **(b)**

Figure 1.32: T_4 vs T_4^D and $T_{4.2}^P$ in (a) and (b) respectively. Notice that the continuous part of the variables has been transformed using the empirical distribution function such that the domain is $[0, 1]$ for all the variables. The transformation allows for an easier assessment of the mutual information between the pairs of variables.

At this point, we have observed that the algorithm rediscovers what we already know to be true. Namely, that the total masses and process durations are well explained by the individual processes. However, a more interesting question is how each of the processes affects each other. More precisely, if we only consider each process without the accumulated random variables T_i, T, M_i, and M, we hope to discover more interesting dependencies between the processes. Also, T_i, etc. can always be computed from the processes, so we do not care for these random variables if we want to understand the dynamics of the system.

Initially, we consider the durations and delays only. Namely, we choose the submatrix of G_{obs} corresponding to the sets T_i^D and T_i^P from Equation 1.13. Still using a symmetric G_{obs}, G_{dir} is computed as shown in Figure 1.33(a). Now, when removing the *trivial* random variables, as they are known to be the sum of others, a clearer image of how the durations and delay of each process depend on each other emerges. A strong association between T_9^P and T_7^D is observed. When plotting the observations of the random variables (again with the continuous part transformed through the empirical density function) in Figure 1.33(b), we indeed observe that for many batches, if the delay for process 7 is non-zero, then the duration of process 9 is greater than if the delay after process 7 was 0. Furthermore, as we have removed transitive effects, it seems as if this association is direct. I.e. something happens during the delay after process 7, the reaction process, which heavily *influences* to the duration of the cooling process 9. Interestingly, from Figure 1.33(a) it does not seem like

T_7^P is associated with T_9^P. In **??** we have shown these two variables plotted against each other and indeed observe that neither is particularly descriptive of the other.



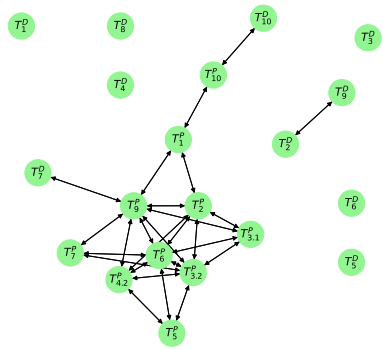
(a) G_{dir} **(b)** T_9^P vs T_7^D

Figure 1.33: When only using the durations and delays, we observe G_{dir} as in (a). A very strong similarity is calculated between T_9^P and T_7^D, which when plotted in (b) is not an unreasonable finding. Namely, if the delay after process 7 is 0, T_9^P is primarily below the 0.6-quantile and otherwise appears to be linearly related to T_7^D.

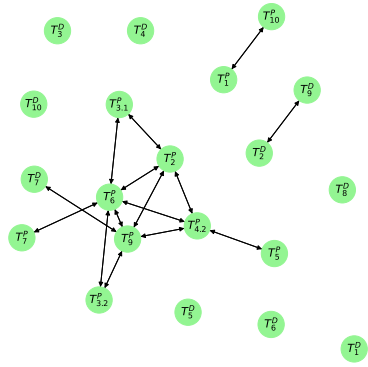
In figure Figure 1.34, we have shown G_{dir} (from Figure 1.33(a)) as a graph with varying thresholds. It is clear that the durations of each process are dependent on each other and depending on the threshold we conclude anything from a simple network structure (Figure 1.34(b) and Figure 1.34(c)) to a much more complicated dependency structure (Figure 1.34(a)) between the durations of the processes. As is also clear from Figure 1.33(a), the delays of the processes are the first random variables we conclude to be independent of both each other and all other process durations except for T_2^D and T_9^D corresponding to the delays when adding material and cooling respectively and the link between T_7^D and T_9^P discussed above.

Like in Subsection 1.3.3, we have a good understanding of the topological structure of the processes. Namely, as the delay of processes is always *after* the process we have that in terms of a topological structure of the random variables, T_i^D is always after T_i^P. For example, T_3^D is always realized *after* $T_{3.1}^P$ and $T_{3.2}^P$ (in that order). Likewise, as shown in **??**, as the processes are executed one by one, the topological structure is a simple chain. This should not be confused with the actual causal structure being a chain as in Section 1.1. As the topolog-

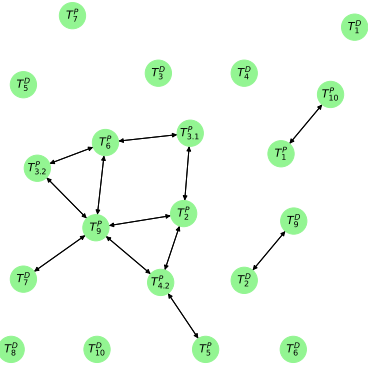
ical structure is a chain, it is also unique i.e. there is exactly one ordering of the random variables. The topological structure is summarized in Equation 1.14



(a) $t = 0.1$



(b) $t = 0.118$



(c) $t = 0.12$

Figure 1.34: G_{dir} from Figure 1.33 represented as a graph for different thresholds t. In general, we observe that the delays appear to be indescribable from the other variables and vice versa. As for the actual durations of the processes, depending on the threshold, we observe a more or less complicated relational structure. Varying the threshold t outside the considered range [0.1, 0.12] does not change the graph noticeably.

$$\begin{aligned} & T_1^P \longrightarrow T_1^D \longrightarrow T_2^P \longrightarrow T_2^D \longrightarrow T_{3.1}^P \longrightarrow T_{3.2}^P \longrightarrow T_3^D \\ \longrightarrow & T_{4.2}^P \longrightarrow T_4^D \longrightarrow T_5^P \longrightarrow T_5^D \longrightarrow T_6^P \longrightarrow T_6^D \longrightarrow T_7^P \\ \longrightarrow & T_7^D \longrightarrow T_8^D \longrightarrow T_9^P \longrightarrow T_9^D \longrightarrow T_{10}^P \longrightarrow T_{10}^D \end{aligned} \tag{1.14}$$

Using this to order the rows and columns of G_{obs} and finally only keeping the upper triangular part, we can deconvolve the network once again, but this time using the topological structure. The resulting G_{dir} is shown in Figure 1.35.

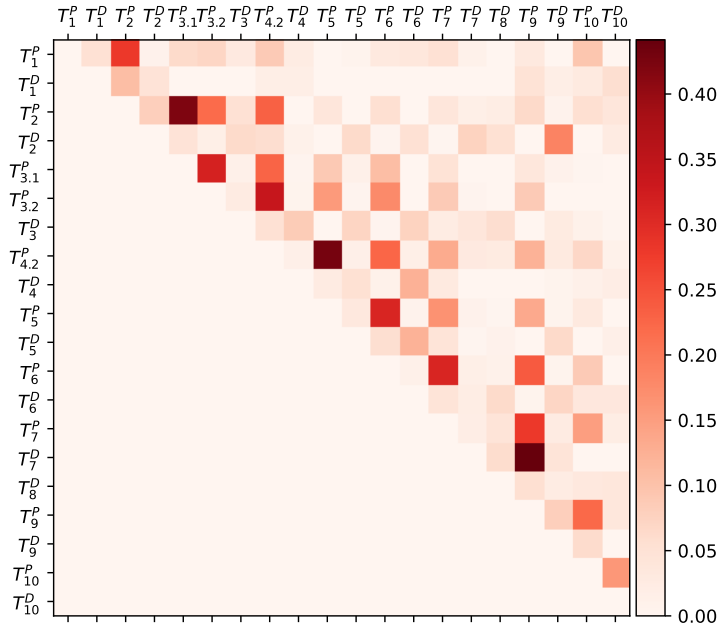
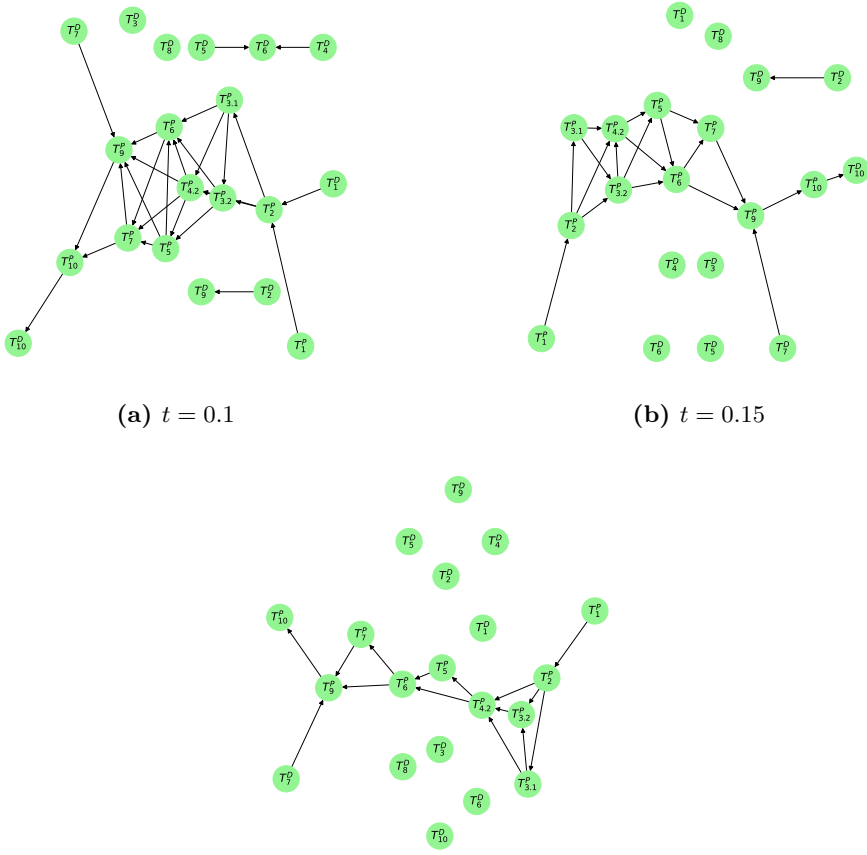


Figure 1.35: G_{dir} based on durations and delays but now with the added assumption of the topological order of the variables. Notice that the order of the variables is different from that of Figure 1.33(a) and hence not easily comparable. We refer to the graphical representations in Figure 1.34 and Figure 1.36 for comparison of dependence and causal structure.

Although it is hard to compare this G_{dir} with the one obtained without the assumption of topological structure in Figure 1.33(a), we see some differences. T_1^P and T_{10}^P are not as directly dependent as originally inferred from the symmetric G_{obs}. The differences are however more noticeable when comparing the graphs

from before with those in Figure 1.36 where the directed G_{obs} has been used instead.



(a) $t = 0.1$

(b) $t = 0.15$

(c) $t = 0.2$

Figure 1.36: Using a triangular G_{obs} resulting in a directed graph, we observe once again that the delays are largely unrelated. This hints that most delays should be treated separately when we only concern ourselves with the duration of processes. An important difference from these graphs to the previous where no topological order was assumed, is that now T_1^P only influences T_{10}^P indirectly where it was a direct link before. In general, we observe that durations only have a direct influence on the next process or the ones after that again. I.e. a more chain-like structure emerges compared to the previous results.

Depending on the chosen threshold t, we observe that still, the delays are the

least predictable from other observations as they are not connected to other random variables. This was also the conclusion from Figure 1.34. The major difference is that now T_1^P influences T_{10}^P indirectly whereas in Figure 1.34 they were always directly connected. Also, a threshold in [0.1, 0.2] seems to result in a good balance between the connectedness and complexity of the graph. In our opinion Figure 1.36(b) obtains a good balance while also making sense in terms of how processes of a production flow, structured as in **??**, could influence later processes.

Finally, we shall try once again to use the levels of the tank after each process along with a topological assumption. However, as it is unclear whether it is the duration of a process that influences the change in levels or the other way around, we shall only make G_{obs} almost triangular. In particular, the entries in G_{obs} related to durations and levels of the same process such as T_1^P and M_1^P are symmetric along the diagonal whilst everything else is removed. This makes G_{obs} *almost* triangular in the sense that it is triangular but with entries in the subdiagonal (in case of an upper triangular G_{obs}).

The resulting G_{obs} can be seen in the appendix, **??**. As we would expect, level changes and durations for the same process are often very related and share much information. Unsurprisingly, $M_{4.3}^P$ does not seem to be related to any other process. This is likely because the process consists of waiting for a control operator. This also holds when deconvolving the network as seen in Figure 1.37 and from the original result in Figure 1.31 where all the random variables were used (although it is harder to see).

When comparing G_{dir} in Figure 1.37 to G_{obs} in **??**, we observe that much of the association between processes originate from indirect effects (as expected) either 1, 2 or 3 processes ago. In particular, the delays, durations, and level changes of process 9 (cooling) appear to be caused or at least explained well by what takes place during process 7 (reaction) which is not unlikely from a chemical point of view. Process 7, in turn, appears to be influenced mostly by process 6 (product transfer).

In Figure 1.38, we have shown G_{dir} represented as a graph with a threshold $t = 0.09$ as this filtered out most small values in G_{dir} while keeping a reasonable structure with only the edges with the most information carried along. As mentioned above $M_{4.3}^P$ is alone. Furthermore, we observe that the delays related to the process 3 (after adding the final solids) and 8 (post-reaction process) are by themselves. The duration of the delay and the change in level are however connected although weakly as per Figure 1.37. This also agrees with Figure 1.36(b) and Figure 1.36(c) where when we only used the durations and delays of the processes, we observed that T_3^D and T_8^D where unconnected nodes. As another example, M_8^P is not connected to any of the other variables invariant to the

choice of threshold t. This also makes sense when comparing to **??**, **??** and **??** where the change in level during the post-reaction (process 8) does not seem to be related to any of the other random variables.

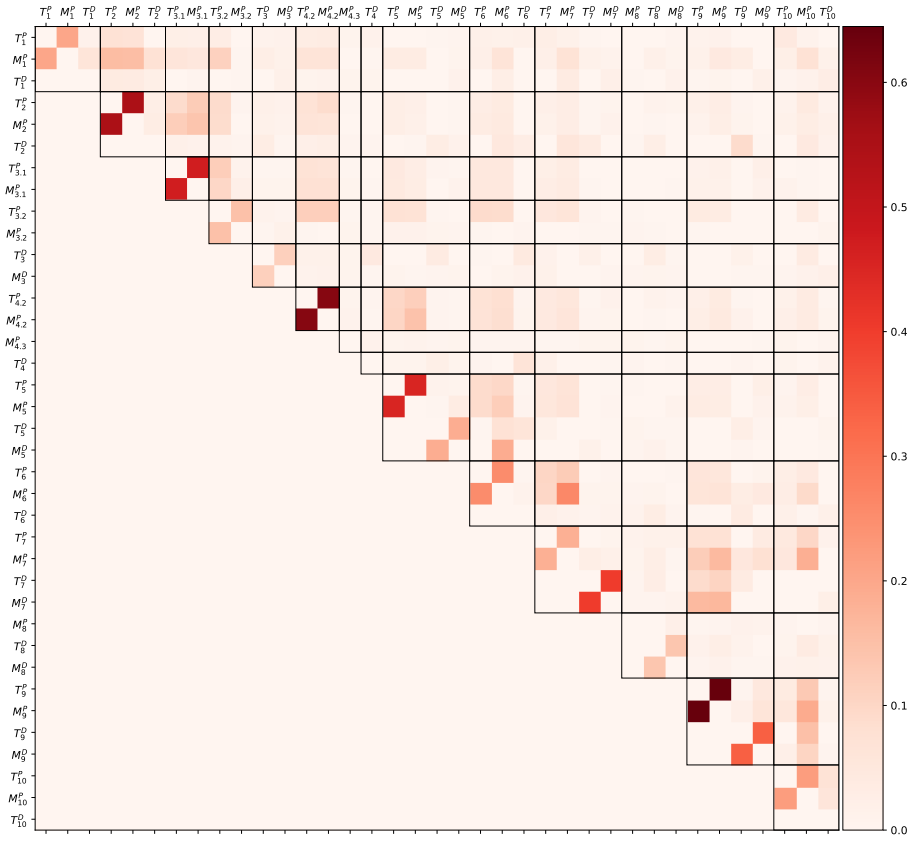


Figure 1.37: G_{dir} from an *almost* upper triangular G_{obs}. Like before, we observe a chain-like structure. However, with the added random variables of changes in levels during a process, the most significant dependencies are between these and the corresponding duration.

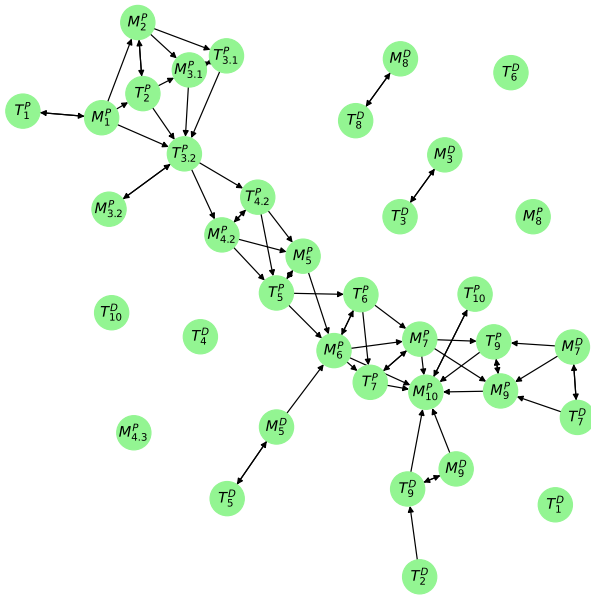


Figure 1.38: G_{dir} represented as a graph using a threshold $t = 0.09$. Once again, we observe a chain-like structure, but the added random variables corresponding to changes in levels, $T_{3.2}^P$ becomes a bottleneck, in the sense most of the behavior of the production system after process 3.2 is irrelevant when knowing the duration of process 3.2. This could also make sense from a practical point of view as the duration of the stirring contains much-combined information of the initial processes and hence is a good descriptor of the behavior of the processes later on.

Other interesting observations can be made from the graph depending on the interests of the reader. However, we round of our discussion of the pharmaceutical data by noting that we have obtained a causal structure for the random variables that in many ways reflect both our understanding of the production layout from **??** and our observations of data. In particular, we observe a relatively simple structure, where many of the processes are only affected by the previous process. Notable deviations of this are the initial processes 1, 2, and 3.1 where raw materials are added. Each of these processes seems to have an influence on the duration of process 3.2 (the stirring). This also makes sense as the more material is added to the tank, the longer it probably needs to be agitated to ensure a consistent mixture of materials to allow for chemical reactions.

Finally, the change in level for process 10 (the transfer of material), appears to be influenced primarily by processes 7 (the chemical reaction) and 9 (cooling) while the duration of process 10 only seems to be related to the change in level and no other process directly. This would make sense practically, as we would think that it is the amount of material that is really influenced by the other processes and the duration of transferring the material is only as needed to be in order to remove the material produced.