Causal discovery and inference of data from production flows

Jonas Bruun Hubrechts

Kongens Lyngby 2024

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

The goal of the thesis is to ...

Summary (Danish)

Målet for denne afhandling er at ...

Preface

This thesis was prepared at DTU Compute in fulfilment of the requirements for acquiring an M.Sc. in Engineering.

The thesis deals with ...

The thesis consists of ...

Lyngby, 31-July-2024

Not Real

Jonas Bruun Hubrechts

Acknowledgements

I would like to thank my....

Contents

Chapter 1

Problemformulering / Introduktion

In many production facilities, planning is a big part of maximizing some index. Whether this is production throughput over some time period and thus often also the economic surplus or some other key index, it is of great importance to have an underlying model to describe the observed variation. In particular in operational research, the schedules may drift in suboptimal ways if the variation is not considered.

Furthermore, from a salesman point of view, expected production and time intervals can be of great use when planning and also building production facilities. Namely, one might find that increasing the volume or efficiency of some part of the facility would increase the production throughput and profitability. This is also known as bottleneck analysis and require some understanding of the underlying mechanics and a stochastic model of this could improve the strength of such results.

Therefore, the primary objective of this paper/thesis is to investigate and model the yield and time of a production flow with focus on the pharmaceutical and chemical production industry. More precisely, we will be building a statistical model for a single process, with the purpose of being able to describe the variation in the yield of the production cycle and production times. This will then be used to analyze potential bottlenecks.

Furthermore, it will be interesting to construct a network of such processes as is typically the case in industry. We shall see how much can be said about such a network and what obstacles one may encounter when trying to analyze such networks which is this thesis will initially be treated as networks of queues.

Chapter 2

Data

In this section, we will describe and analyze the data used in this thesis. The data stems from a simulation study on a chemical batch production system and comprises six time series from which the time stamps (in hours) and level in a tank is of the most interest. The goal of this section is to describe how different phases of the production covary. The data is chosen as it resembles an actual production data set but is also implemented with what at first glance seem to be fairly realistic variation and noise in measurements. Overall, the point of this example data is to exemplify what one may encounter in a real batch production system and from this try to build a model in order to predict or quantify the behavior of the system or learn hidden (causal) structure important for optimization etc.

We define a set of phases/units $\mathcal{U}$ that each batch comprises. In this case, units are identified with IDs 1 through 10 (with subunits such as 3.1) described further in Table 2.2. Then, for each unit, we define the stochastic variables X_u and X_u^D to be the duration and delay after a unit respectively. It is also important to keep track of the level in the tank after each unit is finished, and we thus define variables M_u and M_u^D to be the level in the tank after unit u and its associated delay respectively. As the units are executed in sequence, as is the case in this data, a simple representation of the variables can easily be visualized and is shown in Figure 2.1 below.

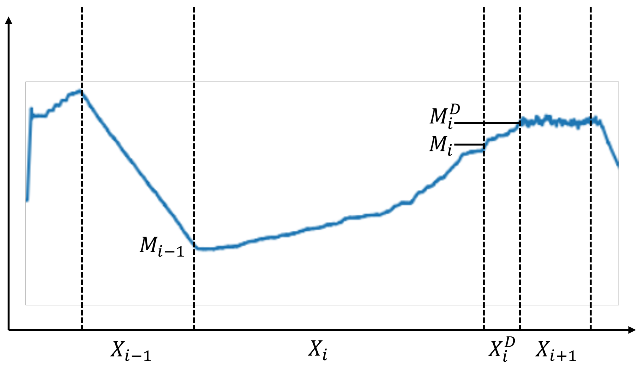


Figure 2.1: Exemplification of the variables X_i, X_i^D, M_i, M_i^D.

2.1 Basic statistics

Initially, we present some basic statistics in Table 2.1 for batches and proceed to discuss the nature of the batches, removing outliers and faulty observations etc.

Cycle	#batch	μ	σ^2	σ	σ/μ
A	66	14.776	3.641	1.908	0.1291
B	64	15.644	3.915	1.979	0.1265
C	61	17.714	2.330	1.526	0.08617
D	60	18.069	6.922	2.631	0.1456
E	60	18.088	9.613	3.100	0.1714
F	63	17.227	7.766	2.787	0.1618

Table 2.1: Per cycle batch duration statistics

Each batch comprises several states. These include adding materials (IDs 1 through 4), centrifugation (ID 5), product transfer (the precipitate generated from the centrifugation, ID 6), chemical reaction (ID 7), a post operation state (Probably to let it cool down to a point where it is ready for further processing, ID 8), Cooling of the product (ID 9), material transfer (transfer the gained product before cleaning of the reaction vessel and/or prepare for the next reaction batch, ID 10). Notice that there is a total of 374 batches throughout the 6 observed cycles.

2.2 Incompleteness on trailing batches

As it may be of interest to investigate the correlation structure of different metrics and variables later on, it is important to understand how each of the batches across the cycles behave. Initially, when looking through the dataset, we observe a few negative phase IDs which will need investigation. However, before we do so, we check that each of the batches actually go through all the states mentioned in [1]. Thus, we take the absolute value of the negative phase IDs to ease the analysis prior to the analysis of the negative phase IDs. After this is done, we observe that not all batches go through all the phases and that some seem to have extra phases not described by [1]. Namely, from Table 2.2, we see that IDs 3 and 4 (which are not described in [1]) have significantly fewer batches going through this phase. But perhaps even more interesting is the final 4 phases where almost all batches goes through these phases.

ID	Count	Description
1.0	374	Addition of liquid raw material `Educt1`
2.0	374	Addition of liquid raw material `Educt2`
3.0	181	
3.1	374	Addition of liquid raw material `Educt3`
3.2	374	Agitation
4.0	163	
4.1	374	Waiting for field operation
4.2	374	Addition of solids
4.3	374	Waiting for control operator
5.0	374	centrifugation
6.0	374	Product transfer
7.0	370	Reaction
8.0	369	Post reaction
9.0	369	Cooling
10.0	368	Material transfer

Table 2.2: The number of batches across all cycles that contains at least one observation for each different absolute phase ID.

Investigating when these inadequacies occur reveals that they are the last batch from each of the cycles. For example, the final batch from cycle A only goes to phase 6 (the product transfer). This can however be explained from the fact that simulation only last for 1100 hours for each cycle and is thus simply cut-off here. As we do not know if these final operations were done at the time the simulations were cut off (which is likely not the case), the final phase for each of the final batches should be disregarded. The cut-off can also be observed in Figure 2.2. Furthermore, throwing away 6 incomplete batches out of the total

374 will likely not harm the analysis and is thus thrown away as this will make the analysis much simpler later on.

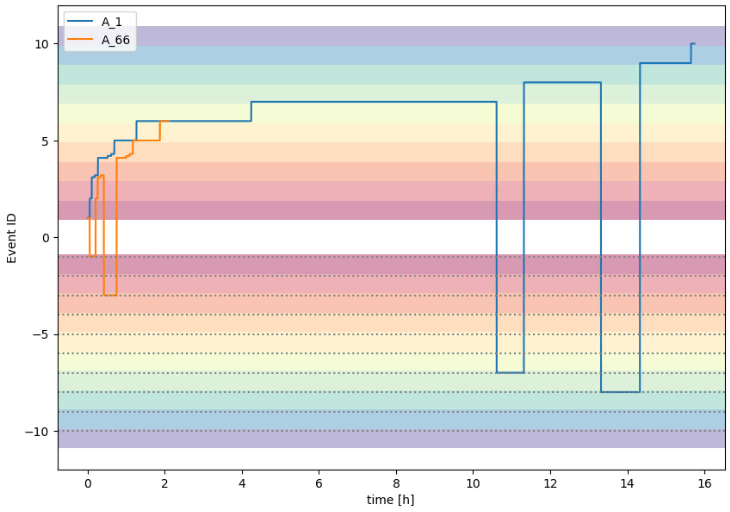


Figure 2.2: The first and last batch from cycle A. It is clear that the final batch is cut-off even before the current phase it finished.

After cutting of the final 6 batches, we have a total of 368 batches of which each goes through all the phases. We thus proceed to discuss the negative phase IDs in the following section, where we also discuss the first four phases.

2.3 Production phases

This part of the process corresponds to events tagged with ID 1 through 10 but will initially concern itself with ID 1 through 4 as much can be learned from the data set here. In Figure 2.3 an example of how the process evolves over time through the different phases is shown. Immediately, we observe something weird, namely the negative event IDs.

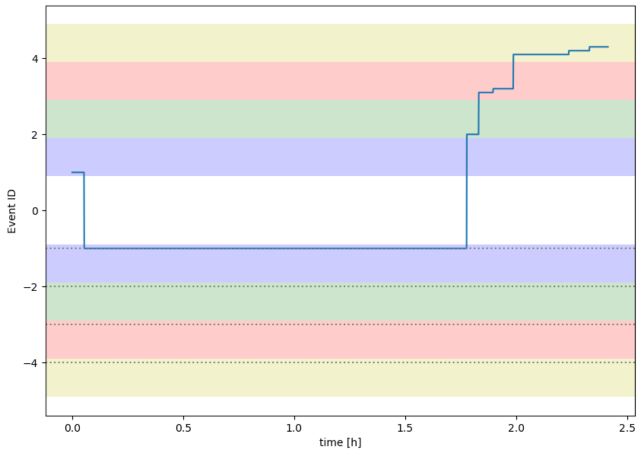


Figure 2.3

To see what is going on here, from data we can see that negative values occur throughout all the six cycles. More specifically, for each negative phase observed, we log in which cycles this occurs. The result is shown in Table 2.3. Notice that -4.1, -4.2 and -4.3 only show up in cycle F, which from [1] is known to be the only one with wrongly labelled phases. We thus suspect that this is indeed the case for these labels and might just have supposed to be the original 4.1, 4.2 and 4.3. To see if nothing funny goes on with these values, these batches are plotted as in Figure 2.3 in Figure 2.4.

Figure 2.4 shows that nothing weird is going on except for the negation of the sub phase's ID. The same can be said for the remaining of the cases where phase ID -4.1, -4.2 and/or -4.3 is used. We thus conclude that these may simply be wrongly labelled thus we convert every such instance to its absolute value and continue with this modified data set from this point on.

Cycle Event	A	B	C	D	E	F
-1	■	■	■	■		
-2				■	■	■
-3	■		■	■	■	■
-4		■	■	■	■	
-4.1						■
-4.2						■
-4.3						■
-5	■	■	■	■	■	■
-6		■	■	■	■	■
-7	■		■	■	■	■
-8	■		■	■	■	■
-9				■	■	■
-10		■		■	■	■

Table 2.3: Occurrences of negative phases IDs. It is observed that sub phases 4.1, 4.2, 4.3 only occur in cycle F which is known to be the only cycle with wrongly labelled phases.

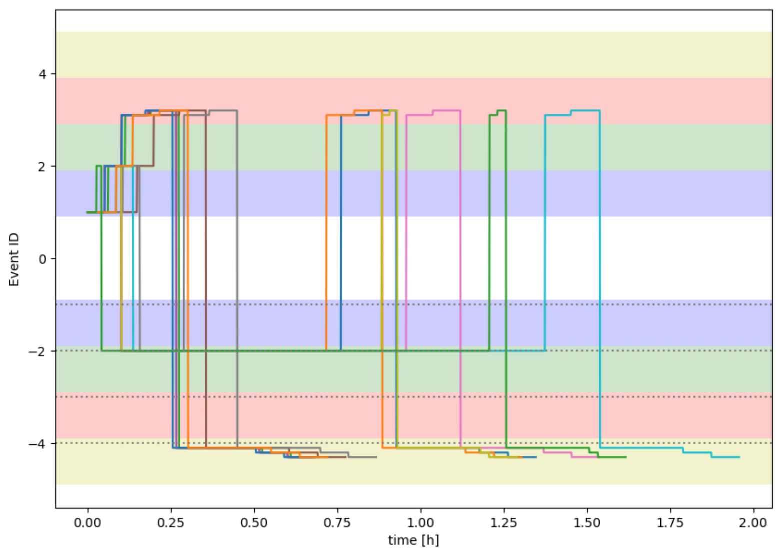


Figure 2.4: 13 of the 48 batches with at least one of the sub phases 4.1, 4.2 4.3 negative.

Having converted the above sub phase IDs we summarize the current situation in regard to negative phase IDs in the following table, Table 2.4. Now all the remaining occurrences of negative phase IDS does not seem to exhibit any structure from looking at Table 2.4. We thus proceed to understand what is going on with the remaining negative phase IDs.

Cycle / Event	A	B	C	D	E	F
-1	■	■	■	■		
-2				■	■	■
-3	■		■	■	■	■
-4		■	■	■	■	
-5	■	■	■	■	■	■
-6		■	■	■	■	■
-7	■		■	■	■	■
-8	■		■	■	■	■
-9				■	■	■
-10		■		■	■	■

Table 2.4: Occurrences of negative phases IDs. It is observed that sub phases 4.1, 4.2, 4.3 only occur in cycle F which is known to be the only cycle with wrongly labelled phases.

When plotting different batches, it is clear that the negative phase IDs only occur at the end of a phase e.g. -1 only happens after 1 and so on. This together with the fact that -3 and -4 also only happen after 3.1, 3.2 and 4.1, 4.2, 4.3 respectively (and we never see a phase labelled 3 or 4) indicate that the negative phase IDs could very well correspond to delays at the end of a phase, which both makes sense from a production point of view but also from [1] where they note that all simulated cycles have been implemented with delays.

At this point, it would seem that the labels of the processes are understood for phases 1 through 10 corresponding to the actual production in each batch. Thus, we proceed by searching relationships and otherwise quantifying the durations of each phase, both delays and duration for each of the phases. As a beginning, histograms for each of the phases and delays are plotted in Figure 2.5. Notice that phases 4.1, 4.3 and 8 are not shown, this is because they always last 15 min, 5 min and 2 hours respectively with the only derivation being in machine precision either when loaded or during calculations. Furthermore, notice that for the negative IDs i.e. the delays, the orange bar. This bar represents the cases where no delay was observed which is thus modelled as an atom at 1.

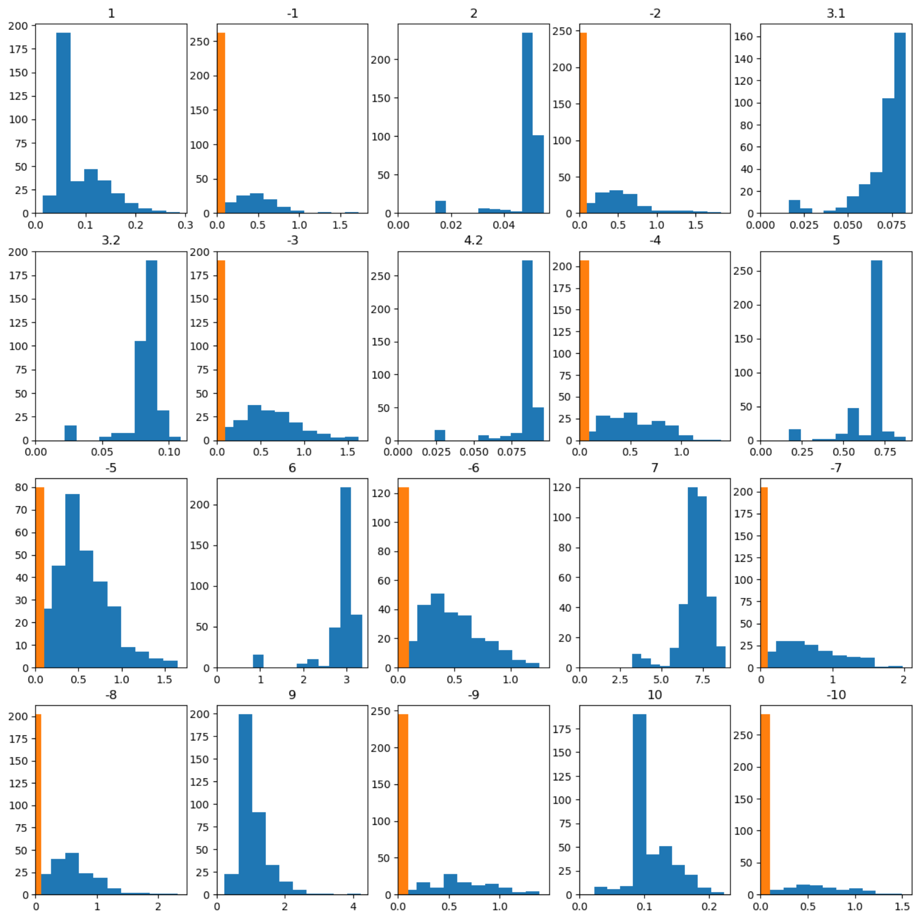


Figure 2.5: Histograms of all phases and delays which are non-constant.

Apart from the above comments, not much catches the eye when looking at Figure 2.5, and we thus proceed by checking if any correlation is immediately present.

2.3.1 Correlations

Lige en korrelationsmatrix

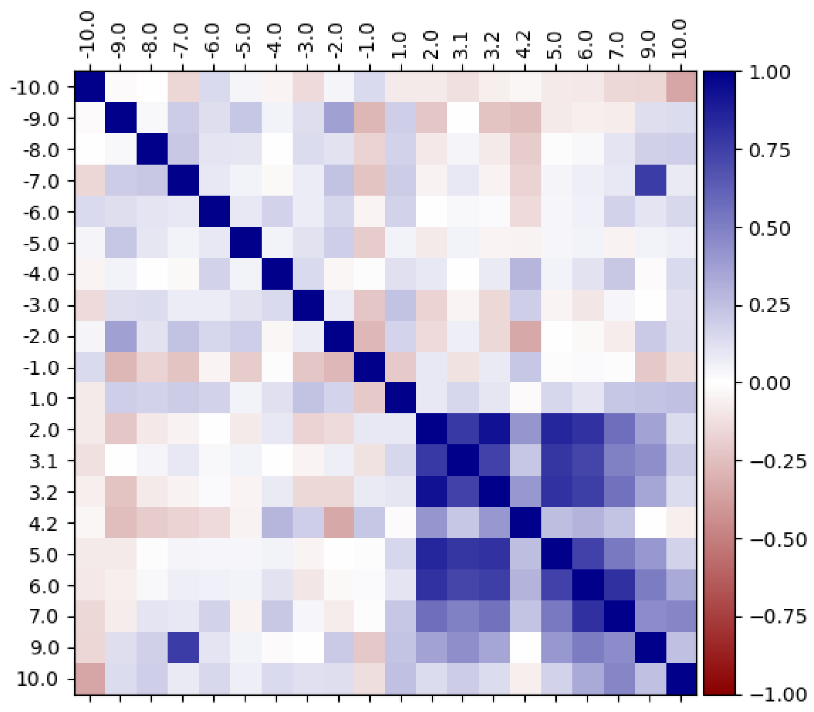


Figure 2.6: Correlation matrix for all phases with non-constant duration.

9 og 10 er ikke specielt korreleret med noget (afkøling og materiale overførsel). Ellers er 2 fremt il og med reaktionen alle korrelerede med hinanden. Eftersom rent fysisk det udvikles i tid, må handlingen i 2 påvirke de næste osv.

Umiddelbart lidt spøjst hvis delay på 7 (reaktion) skulle have noget med tiden for afkøling at gøre, især at den skulle være positiv (ville man ikke tro delay efter produktion ville afkøle mere og dermed reducere behov for afkøling, medmindre varmt steam bliver tilføjet også under delay på 7)

Herunder er samme korrelationsmatrix, dog hvor delay og phasens varighed lagt sammen (også med sub phases såsom 3.1, 3.2 og -3 tilsammen bliver 3)

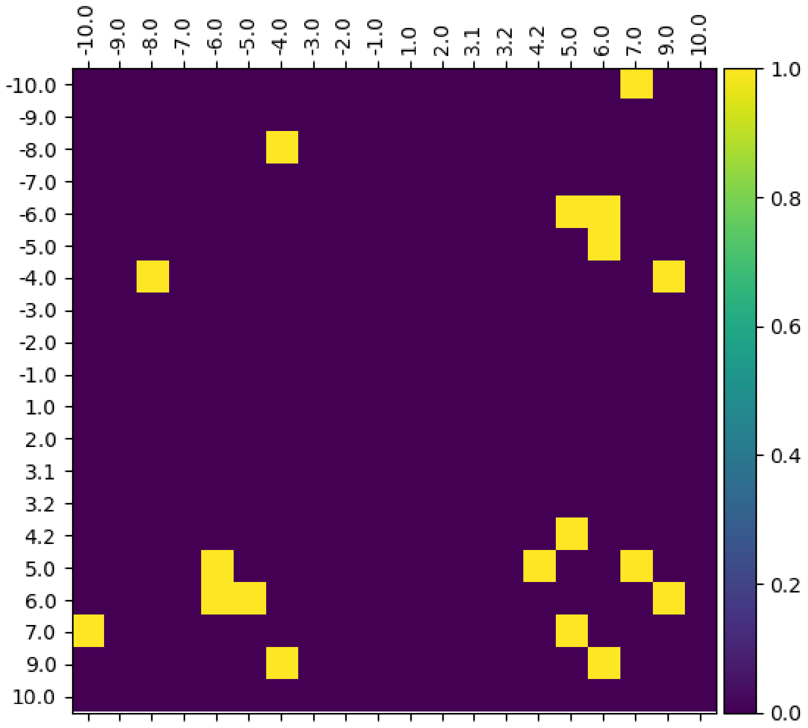


Figure 2.7: Permuteringstest med $\alpha = 0.05$. Also run with less simulations but same result at 1 mil and 10k sims. The Benjamin-Hochberg procedure on the upper (or lower) triangle reveals that none of the correlations are significant.

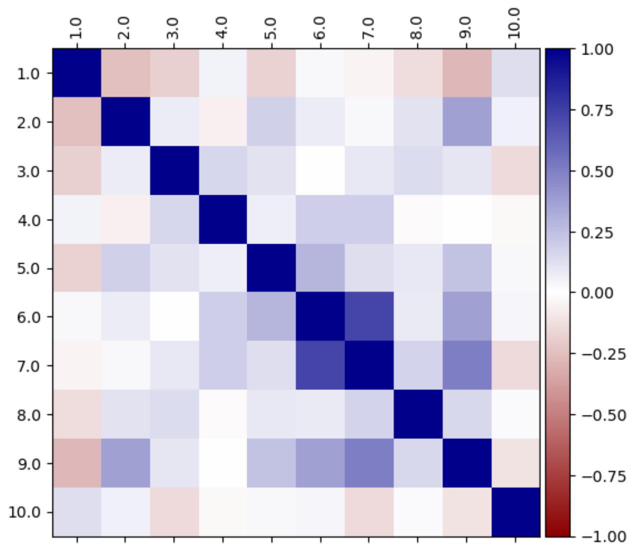


Figure 2.8: Correlation matrix for all phases collapsed

Ligeledes scatter plots for superdiagonalen i ovenstående matrix. Altså phase 1 overfor phase 2, phase 2 overfor phase 3 osv. Er farvelagt efter hvilken cycle de kommer fra. Table 2.4 forklarer hvorfor nogle af de horisontale fremkommer sammen med Figure 2.5 (selve produktionstiden er ret kort sammenlignet med delay.)

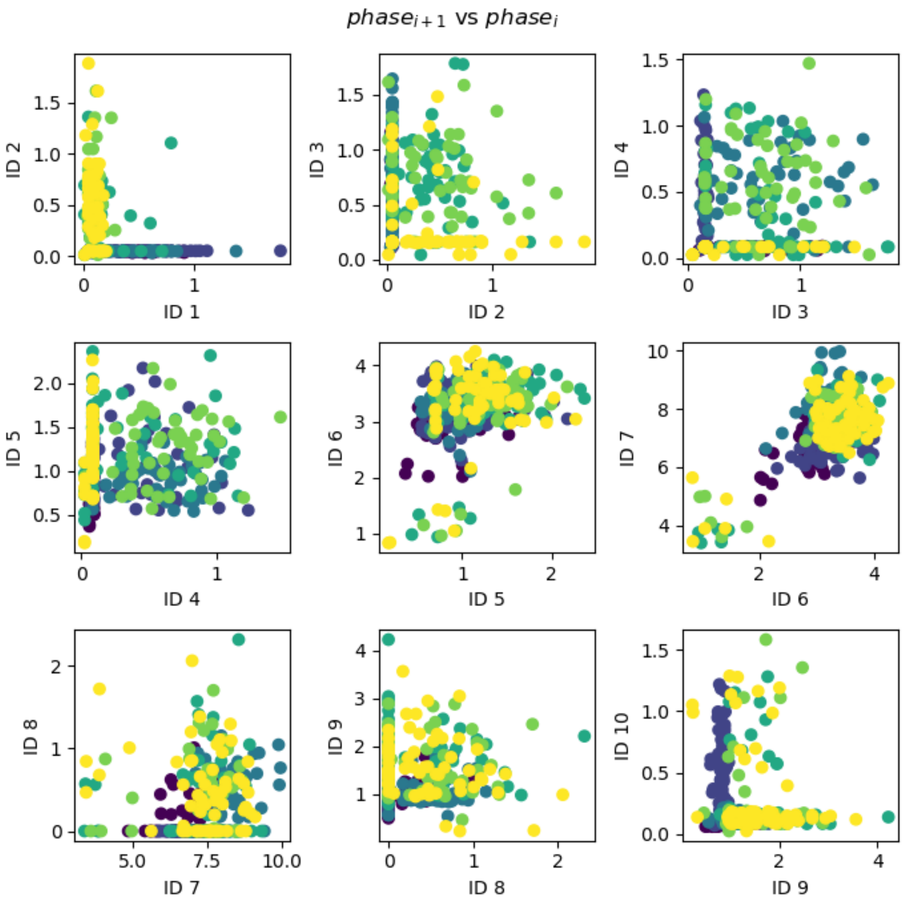


Figure 2.9: Phases vs their next phase when collecting everything regarding a single phase into a total time duration

2.4 Cleaning operations

Sometimes, the vessel is cleansed. This is however not every time after a batch so might be interesting to investigate further. Initially, per cycle, the cleanings are summarized in the following table with basic statistics. As can be seen, there is quite some differences.

The most notifiable differences per batch are the number of cleanses especially when comparing to Table 2.1. For the first two cycles, the cleanses seem to be in between every batch, which is indeed also the while the later four are only sometimes. Furthermore, although the cleanses are between every batch for cycles A and B, the variances are extremely different. For the last four cycles, they seem to be grouped further, E and F are very alike while cleanses in C and D are generally longer although D has a substantially smaller variance than C.

Cycle	#ops	min	max	μ	σ^2	σ	σ/μ
A	65	1.113	3.067	1.917	0.269	0.518	0.270
B	63	1.324	1.751	1.566	0.00883	0.0939	0.0600
C	9	1.544	3.306	2.153	0.277	0.526	0.245
D	10	1.474	2.009	1.581	0.0212	0.146	0.0922
E	10	0.827	1.584	1.465	0.0462	0.215	0.147
F	10	0.748	1.610	1.466	0.0595	0.244	0.166

Table 2.5: Per cycle cleansing statistics

Cycle	A	B	C	D	E	F
$\mathbb{E}\,[\widehat{\sum X_u}]$	13.993	13.898	15.343	14.471	14.589	14.418
$\text{Var}\,(\widehat{\sum X_u})$	0.95636	0.46587	0.76111	4.9589	4.2678	5.3545
$\sum \widehat{\text{Var}\,(X_u)}$	0.50590	0.31182	0.36667	1.8322	1.5788	1.9696
$\mathbb{E}\,[\widehat{\sum X_u^D}]$	0.96398	1.9402	2.4503	3.6050	3.7390	3.0041
$\text{Var}\,(\widehat{\sum X_u^D})$	0.31843	0.39117	0.90187	1.2468	1.2787	1.0462
$\sum \widehat{\text{Var}\,(X_u^D)}$	0.34921	0.53198	0.74914	1.4357	1.2454	1.3099
$\mathbb{E}\,[\widehat{\sum X_u\, X_u^D}]$	1.9321	1.5001	4.8191	6.4225	6.0405	6.3343
$\text{Var}\,(\widehat{\sum X_u\, X_u^D})$	3.7798	0.89920	16.870	22.133	12.660	16.194

Table 2.6: Each of the time related variables X_i and D_i and variance description.

To verify these observations and potentially discovering more important facts of their probability distributions, histograms are plotted in the following Figure 2.10. We indeed again observe the likeliness between the cycles A and B, C and D, E and F respectively. Also, for the first two cycles and more so cycle B, the cleaning times are somewhat normally distributed although cycle A has a very heavy right tail in that case. The later four cycles only have 10 observations but the mode (i.e. peak) seem to be about the same.

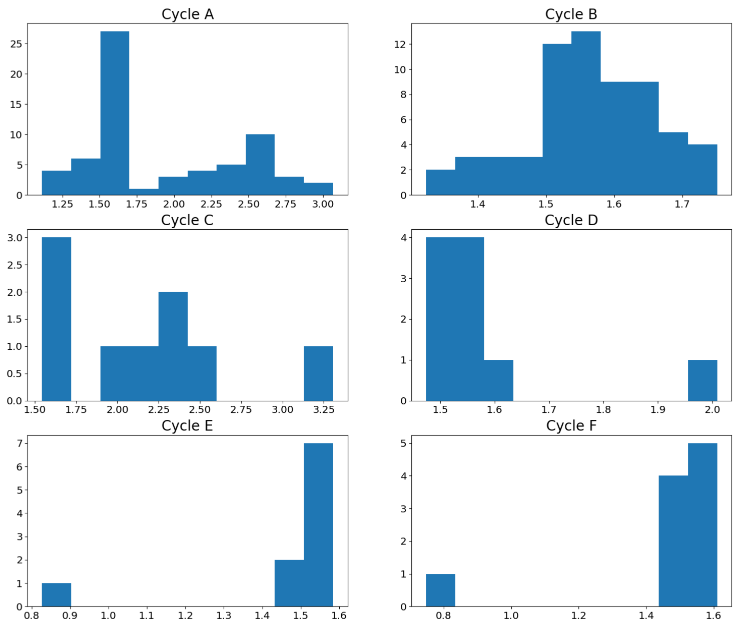


Figure 2.10: Each of the 6 cycles, cleaning operations histograms.

From the above observation of like modes one may want to observe the histogram of the combined set of cleaning times. In particular, under the hypothesis that the durations are actually from the same probability distributions and realized independently within each cycle a histogram of all the observations are of interest and is shown in Figure 2.11 below.

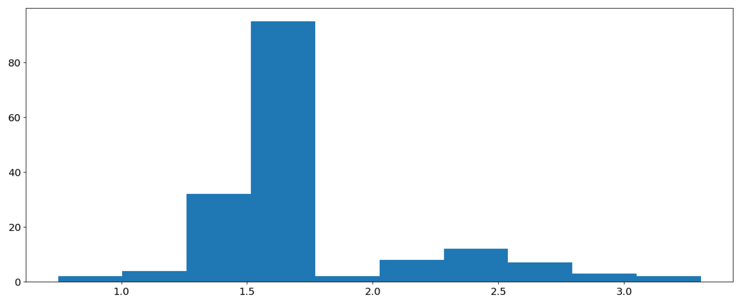


Figure 2.11: Combined cleaning operations histograms.

Finally, to get a better overview of the irregularities is the number of cleaning periods (mostly concerning cycles C through F), each cleaning operation is shown in the following Figure 2.12. The vertical shaded rectangles signify the period in which a cleaning operation is taking place. Furthermore, the event IDs are shown but to get a clearer view on what is going on, a single rectangle (zoomed in) is shown in Figure 2.13.

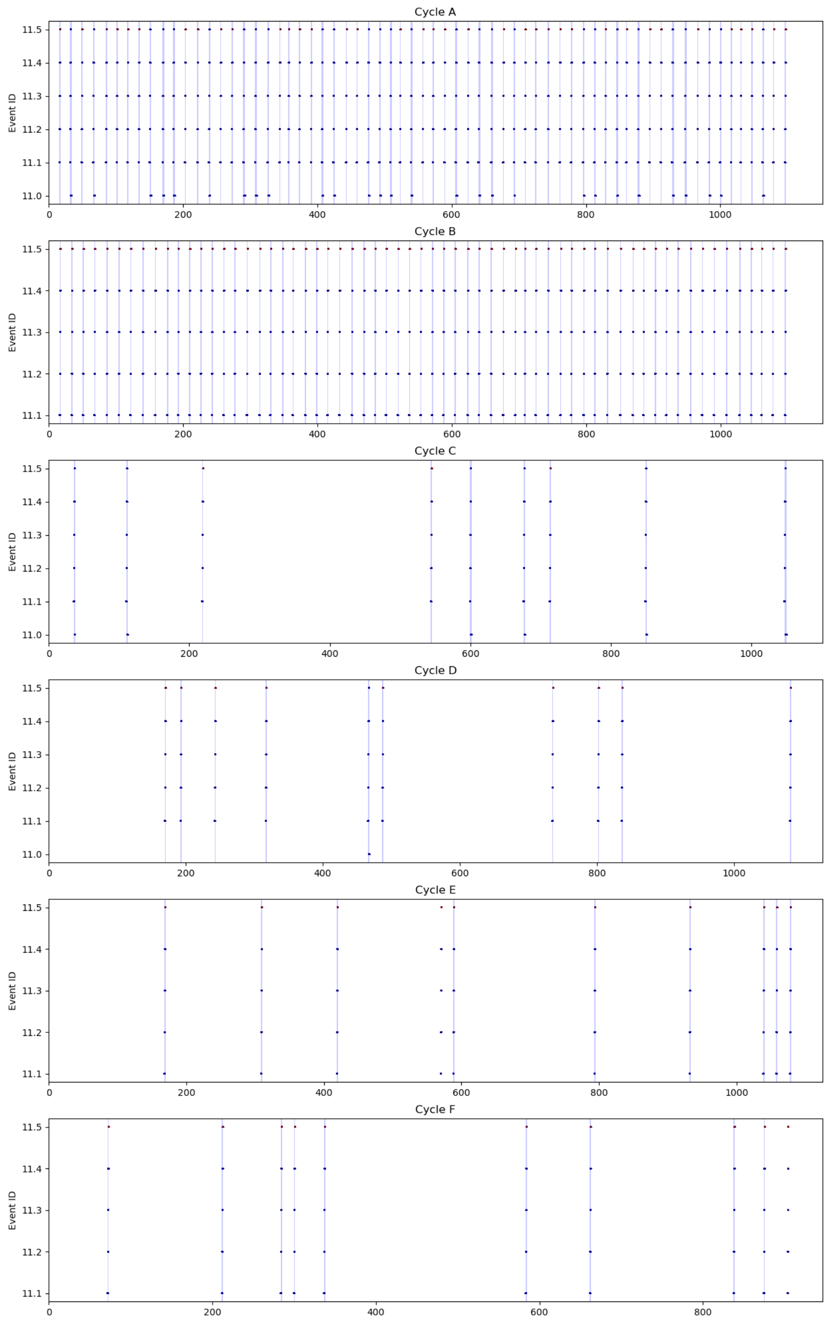


Figure 2.12: Each of the 6 cycles, cleaning (corresponding to `BatchID = 0`). Each *(*Cleaning Procedure), CIP, is highlighted with an opaque interval (the blue rectangles). The dots marked with red (only ID 11.5, but not all of these are red), is if the Cleaning ID is 0.

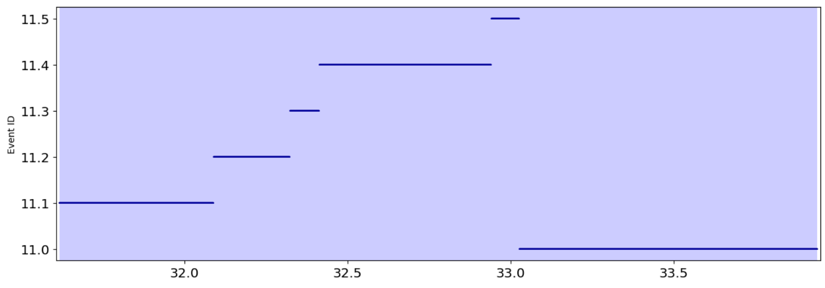


Figure 2.13: A single blue rectangle zoomed in

It is observed that the observations marked with red in figure 2.12 occur exactly when that specific cleaning operation does not go to the state 11.0 after the flush of the tank (event ID 11.5) and vice versa. It is hard to conclude what this may mean, but the cleaning being in state 11.0 may indicate that the system is idle before continuing the next batch like what is observed from the other steps of the process flow. Also, it is noted that while the red dots occur nothing else is happening according to the dataset.

From a modelling point of view, the cycles C through F can be thought of as the cleansing operation having a probability of not happening or equivalently as having a duration of 0. It is thus of interest to observe what the probability of cleaning after an operation is. From Table 2.1 and Table 2.5, we that indeed for cycles A and B, the probability is 100 % when disregarding the possibility of cleaning after the final batch. Hence, we see that for the remaining cycles, the probabilities of cleaning the tank after an operation are as in Table 2.7

Cycle	% cleaning
A	100.00
B	100.00
C	15.00
D	16.95
E	16.95
F	16.13

Table 2.7: Per cycle probability of cleaning

Furthermore, let C_i denote whether the ith batch is followed by a cleaning of the tank or not. It is then of interest if the next batch is followed by a cleaning given whether the current batch is followed by a cleaning. In particular, we count for each of the cycles the transitions which are shown in the following tables. Notice that the number of observations is two less than the total number of batches within each specific cycle. This is due to the last batch is never followed by a cleaning (nor is the first batch superseded by a cleaning procedure) which results in one less observation and also due to the fact that we are logging transitions and hence lose another observation. To test for randomness, a Chi-squared test is carried out on each of the cycles to check for independence. It is observed all the cycles exhibit independence between the groups i.e. there is no statistical evidence for information is gained about if the next batch is followed by a cleaning operation given whether the current batch is followed by a cleaning operation.

C_i \ C_{i+1}	No	Yes
No	41	9
Yes	9	0

(a) C, $p = 0.3293$

C_i \ C_{i+1}	No	Yes
No	41	8
Yes	7	2

(b) D, $p = 0.6456$

C_i \ C_{i+1}	No	Yes
No	41	7
Yes	7	3

(c) E, $p = 0.3532$

C_i \ C_{i+1}	No	Yes
No	41	9
Yes	9	1

(d) F, $p = 1.0000$

Table 2.8: Contingency table for Cycle C-F

Thus collecting the observations from all the last four cycles, we may want to model the atom of the cleaning procedure independently of the previous batch and with a probability of 0.8375 corresponding to the cleaning procedure only being carried out $16,25\%$ of cases.

Finally, we show the autocorrelation function for each the four cycles C-F in Figure 2.14 and note that all the ACF stay within the 95% confidence interval.

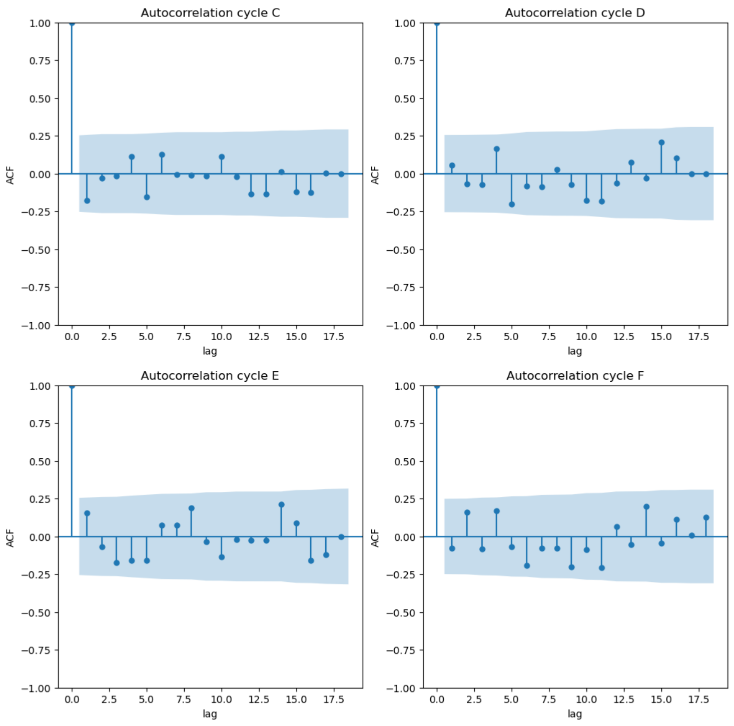


Figure 2.14: Autocorrelation function for each of the final 4 cycles. As can also be seen from this there seem to be no information to be gained of C_i from C_{i-1}.

Chapter 3

Method

General introduction to metodeafsnittet og causalitet

3.1 Causal discovery

In this section, we shall discuss the method for network deconvolution, originally proposed by [2]. The underlying problem is inferring direct effects and dependencies. From this, using prior information on the production setup, we shall be able to infer causal dependencies by directing the resulting edges from the network deconvolution (ND) algorithm. Particularly, the framework and general algorithm proposed by Feizi et al. stems from a graph-theoretic approach to the problem of inferring direct dependencies. Namely, suppose that observations are made of some properties of in this case a chemical process. We shall represent these properties as vertices (nodes) V and dependencies between properties as edges. Initially, when observing the vertices, we observe both direct and indirect effects. Particularly, a vertex v_1 might influence some other vertex v_3 through another vertex v_2 if v_2 depends on v_1 and v_3 of v_2. In this case, we will observe that v_1 influences v_3, but actually it is v_2 that has a direct influence on v_3. In graph-theoretical terms, we thus observe the transitive closure of the information that flows between vertices but want to infer the underlying network structure.

An important note on the algorithm to come is that we only use vertices that we have observed. Namely, the underlying structure might be as in Figure 3.1a with an unobserved node/variable (named U in this case). However, without any more assumptions or modelling choices we would (ideally) infer the network structure depicted in Figure 3.1b. With these initial comments, we proceed

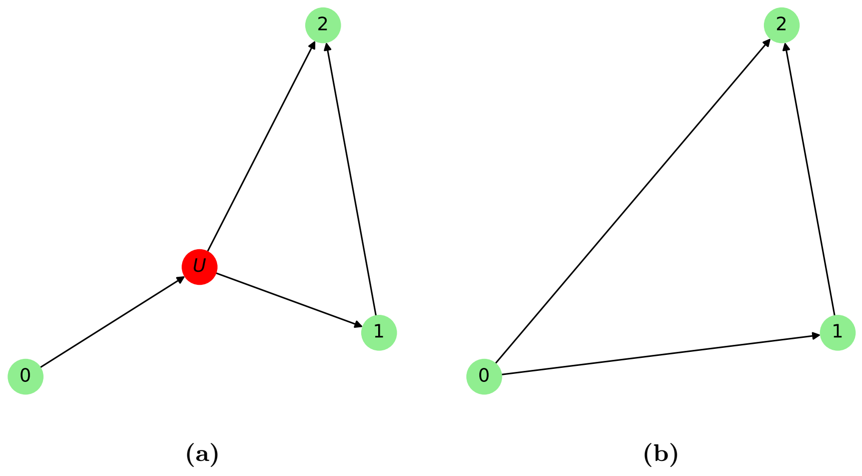


Figure 3.1: (a) An example of a causal structure depicted as a graph. When observing the network, only nodes 0, 1 and 2 are observed/recorded. (b) The resulting inferred graph from observational data. Although this is not a complete picture of the true underlying dynamics of the system, if only the observed variables are of interest, this will be an equally proper representation of the system. Furthermore, in practice this means no further assumptions are made which can and can not be of desire. Namely, if prior information is accessible one might introduce new nodes in the inferred network.

with the general setup and assumptions for network deconvolution based on observations.

3.1.1 Setup and assumptions

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

The method proposed by [2] then works as follows. Given an observed matrix $G_{obs} \in \mathbb{R}^{N \times N}$ of similarities between each pair of variables, we shall deduce a matrix $G_{dir} \in \mathbb{R}^{N \times N}$ of direct similarities between each pair of random variables X_i and X_j. The measure of similarity, can in practice be any desired measure such as correlation or mutual information which we will focus on in this thesis. See Section 3.2 for a further discussion on these two measures and Section 3.3 for how to obtain such a matrix. Note that the algorithm presented will in theory work for non-symmetric measures as well such as *Interaction information*, *Directed information* and *Normalized information*.

The (direct) network is then presented by the discovered G_{dir} containing only the direct effects i.e. interaction between pairs of variables which can be viewed as weights on the edges of the complete graph with nodes representing the random variables. As we shall see in Subsection 3.3.3, the algorithm is somewhat robust to noise in the sense that we can ensure accuracy depending on the level of noise observed present in G_{obs} and on the norm chosen (from a certain, although rather general, set of norms). This hints to that a threshold on the inferred weights on the edges of the network might be a good idea which is further supported by the facts that often only the most influential variables are of importance when trying to control the process.

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

$$G_{obs} = G_{dir} + G_{indir} \tag{3.1}$$

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

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

$$G_{indir} = G_{dir}^2 + G_{dir}^3 + \dots \tag{3.2}$$

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

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

where e_{ij} is the (i,j) element of G_{dir}. Immediately, we observe that e_{ii} is of interest in terms of its physical meaning. The co-dependence between a random variable and itself might be somewhat ambiguous or even undefined depending on the measure. Thus, the notion of (non-existing) edges e_{ii} will be of interest later on when using the method on controlled cases. We note that in G_{obs} we shall in general set these elements to 0.

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

$$G_{obs} = G_{dir} + G_{dir}^2 + G_{dir}^3 + \cdots = G_{dir} + G_{dir}\, G_{obs} \tag{3.3}$$

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

$$G_{obs} = G_{dir}\left(I - G_{dir}\right)^{-1} \tag{3.4}$$

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

$$G_{dir} = G_{obs}\left(I + G_{obs}\right)^{-1} \tag{3.5}$$

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

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

where $\Lambda_{dir} = \Lambda_{obs}\left(I + \Lambda_{obs}\right)^{-1}$ which is also a diagonal matrix.

We note that from the above one needs $\left(I + G_{obs}\right)^{-1}$ to be well-defined which is equivalent to $-1 \notin \sigma_{G_{obs}}$ i.e. -1 is not an eigenvalue of G_{obs}. To see that this is indeed the case whenever $\rho\left(G_{dir}\right) < 1$, and that $I + G_{obs}$ is thus invertible we use Equation 3.4 and simplify

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

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

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

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

As we shall later use some assumptions regarding causality i.e. directing the edges in the graph, we shall investigate the effect of letting G_{obs} be a triangular matrix which corresponds to a directed acyclic graph. Namely, in the following, we show that given the existence of G_{dir} (with necessary and sufficient conditions on G_{obs} as given above), G_{obs} is triangular if and only if G_{dir} is triangular. Thus, by directing the observed similarity (by removing half the edge weights in G_{obs}), we also infer a directed graph G_{dir}.

Clearly, if G_{dir} is triangular, so are the powers G_{dir}^{i} for all $i \in \mathbb{N}$ and hence if the infinite sum $\sum_{i=1}^{\infty} G_{dir}^{i}$ converges, G_{obs} is triangular as well. To show the other way, assume that G_{obs} is triangular and is the result of a G_{dir} with spectral radius smaller than 1. By Equation 3.5, G_{dir} is triangular if the inverse of $I + G_{obs}$ is triangular (upper triangular if G_{obs} is also upper triangular and similarly for lower triangular). This is indeed the case as in general, the inverse of a triangular matrix is also triangular provided that the diagonal elements are non-zero which is true as $I + G_{obs}$ has only ones in the diagonal as we will later assume in Equation 3.8. A simple proof of this is as follows. Without loss of generality, we assume that a matrix T is upper triangular. Let D be the diagonal elements of T and T_u be the remaining strictly upper triangular part of T such that $T = D + T_u$. Then, assuming that D has non-zero diagonal elements, $T = D\left(I + D^{-1}T_u\right)$ and hence, we have that

$$\begin{aligned} T^{-1} &= \left(I + D^{-1}T_u\right)^{-1} D^{-1} \\ &= \sum_{i=0}^{\infty} \left(-D^{-1}T_u\right)^{i} D^{-1} \end{aligned}$$

which is clearly also upper triangular. Thus, we conclude that G_{obs} is triangular if and only if G_{dir} is (under the assumption G_{dir} exists).

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

3.2 Information measures and computation

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

3.2.1 Copula

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

$$\begin{aligned} F(x_1, \ldots, x_N) &= \mathbb{P}\left(X_1 \leq x_1, \ldots, X_N \leq x_N\right) \\ &= \mathbb{P}\left(F_1\left(X_1\right) \leq F_1\left(x_1\right), \ldots, F_d\left(X_d\right) \leq F_d\left(x_d\right)\right) \\ &= C\left(F_1\left(x_1\right), \ldots, F_N\left(x_N\right)\right) \end{aligned}$$

Letting $u_i = F_i\left(x_i\right) \in [0,1]$ it is clear that C is a distribution function as described above [3]. Furthermore, it follows that the marginals of C are uniform as $F_i\left(X_i\right)$ is uniformly distributed. We thus define a copula in probabilistic terms as

Definition 3.1 (Copula). *A function $C : [0,1]^d \to [0,1]$ is called a copula if it has uniform marginals and is a distribution function for a d-dimensional random vector $\mathbf{X}$.*

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

Theorem 3.2 (Sklar's theorem). *For a random vector $\boldsymbol{X}$ with CDF F and univariate marginal CDFs $F_1, \ldots, F_d$. There exists a copula C such that*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)) \tag{3.6}$$

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

Note that the last statement for non-continuous random variables can be made unique by instead using subcopulas, a generalization of copulas with domain I only a subdomain of the unit hypercube $\mathbb{I}^d = [0,1]^d$ containing all faces of the unit hyper cube. However, there are infinitely many ways of extending such a subcopula to a copula C[4]. In our case, this means that for discrete and/or mixed variables, we will later have to work around this non-uniqueness when calculating mutual information. The example made by Geenens[4] is a bivariate random vector of independent variables $X \sim \text{Bern}(\pi_X)$ and $Y \sim \text{Bern}(\pi_Y)$. The support of F_X and F_Y is then $\{0, 1-\pi_X\}$ and $\{0, 1-\pi_Y\}$ respectively. Due to the restriction on the boundary of the unit square, the only unique point of a copula C is then $(1-\pi_X, 1-\pi_Y)$, and by independence we must have

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

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

From Equation 3.6 we see that a copula is thus simply just a function that *couples* the marginals of a random vector to the joint distribution. The following corollary follows immediately

Corollary 3.2.1 (Coordinate transformation)**.** *Under the assumptions of Theorem 3.2, given any set $(T_1, \dots, T_d)$ of strictly increasing functions, if C is a copula of $(X_1, \dots, X_d)$ then it is also a copula of $(T_1(X_1), \dots, T_d(X_d))$.*

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

$$\begin{aligned} F_{Y_i}(y_i) &= \mathbb{P}(Y_i \le y_i) \\ &= \mathbb{P}\left(T_i^{-1}(Y_i) \le T_i^{-1}(y_i)\right) \\ &= \mathbb{P}\left(X_i \le T_i^{-1}(y_i)\right) \\ &= F_{X_i}\left(T_i^{-1}(y_i)\right) \end{aligned}$$

The above is easily generalized for a joint distribution as well. Thus, by the existence of a Copula C for $\boldsymbol{X}$

$$\begin{aligned} F_{\boldsymbol{Y}}(y_1, \dots, y_d) &= F_{\boldsymbol{X}}\left(T_1^{-1}(y_1), \dots, T_d^{-1}(y_d)\right) \\ &= C\left(F_{X_1}(T_1^{-1}(y_1)), \dots, F_{X_d}(T_d^{-1}(y_d))\right) \\ &= C\left(F_{Y_1}(y_1), \dots, F_{Y_d}(y_d)\right) \end{aligned}$$

where Sklar's theorem have been used for the second equality. The above shows that C is indeed also a Copula for $\boldsymbol{Y} = (T_1(X_1), \ldots, T_d(X_d))$. $\square$

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

Proposition 3.3. *Since T_i is strictly increasing, the inverse T_i^{-1} exists and is also strictly increasing. Thus, the above implication is bidirectional and hence for strictly increasing functions T_i, C is a copula of $(X_1, \ldots, X_d)$ if and only if it is a copula of $(T_1(X_1), \ldots, T_d(X_d))$.*

3.2.2 Mutual information and Copula entropy

In this section we introduce Copula entropy as done in [5] and see how it actually is equal to the well known mutual information (multiplied by -1) and hence as a corollary that mutual information is independent of marginals. The name comes from the general definition of (differential) entropy as we shall see shortly. However, first we define mutual information between a set of random variables

Definition 3.4 (Mutual information). *For a random vector $\boldsymbol{X} = \{X_i\}$, we define the mutual information as*

$$I(\boldsymbol{X}) = \mathbb{E}\left[\log_b\left(\frac{f(\boldsymbol{X})}{\prod_i f_i(X_i)}\right)\right]$$

where $\mathcal{X}$ is the domain of the random vector $\boldsymbol{X}$. The base of the logarithm b is often chosen to be 2, e or 10 although the choice is unimportant as all logarithms are equivalent up to a scaling factor.

We note that later on, as the choice of b will result in a scaling of G_{obs}, but we will also introduce a scaling parameter α for G_{obs} to both ensure the convergence of the algorithm and to control higher order effects, we shall in general choose $b = e$.

An important property of mutual information is that the continuous version is the limit of the discrete mutual information for random (continuous) vector discretized as the mesh size goes to zero i.e. recovering the continuity of the random vector. This is discussed in Subsection 3.2.3. For now, we proceed with the definition of (joint) entropy for both discrete and continuous random vectors.

Definition 3.5 (Entropy)**.** *The (joint) entropy of a random vector $\boldsymbol{X}$ is defined as*

$$H(\boldsymbol{X}) = -\mathbb{E}\left[\log_b f(\boldsymbol{X})\right]$$

Where, often for continuous random vectors, we denote the (differential) entropy as $h(\boldsymbol{X})$ instead.

We note the need for two separate notations of entropy as differential entropy is not the limit of (discrete) entropy in the way mutual information is. Again, this is further discussed in Subsection 3.2.3.

Before discussing Copula entropy (CE), we note a very useful relation between entropy and mutual information. Indeed, we shall later use this to show that mutual information in the continuous version is the limit of the discretization.

Lemma 3.6 (Mutual information and entropy relation)**.** *For a continuous random vector $\boldsymbol{X}$, the (joint) mutual information $I(\boldsymbol{X})$ can be decomposed into a sum of differential entropies as*

$$I(\boldsymbol{X}) = \sum_{i=1}^{d} h(X_i) - h(\boldsymbol{X})$$

where d is the dimension of $\boldsymbol{X}$. The same is true for discrete variables but with entropy H instead of differential entropy h.

Proof. This follows immediately from the definition of mutual information and entropy:

$$\mathbb{E}\left[\log_b \frac{f(\boldsymbol{X})}{\prod_i f_i(X_i)}\right] = \mathbb{E}\left[f(\boldsymbol{X})\right] - \sum_i \mathbb{E}\left[f_i(X_i)\right]$$

□

With the definitions of mutual information and entropy we are finally ready to introduce Copula entropy.

Definition 3.7 (Copula entropy)**.** *For a continuous random vector $\boldsymbol{X}$ with a uniquely defined Copula C, and Copula density c, we define the Copula entropy CE of $\boldsymbol{X}$ as*

$$CE(\boldsymbol{X}) = h(\boldsymbol{U})$$

where $\boldsymbol{U}$ has density c. In particular,

$$CE\left(\boldsymbol{X}\right) = -\mathbb{E}\left[\log_b c\right]$$

As stated above, Copula entropy is actually equal to the negative mutual information which we state as a theorem

Theorem 3.8 (Equality of Copula entropy)**.** *For a continuous random vector $\boldsymbol{X}$, the Copula entropy CE is equal to the negative joint mutual information of $\boldsymbol{X}$*

$$CE\left(\boldsymbol{X}\right) = -I\left(\boldsymbol{X}\right)$$

Proof. By Theorem 3.2, letting $x_i = F_i^{-1}\left(u_i\right)$, we can relate the copula density to the joint density of $\boldsymbol{X}$ and its marginals

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

It follows directly that

$$\begin{aligned}
-CE\left(\boldsymbol{X}\right) &= \int_{[0,1]^d} c\left(\boldsymbol{u}\right) \log c\left(\boldsymbol{u}\right)\, d\boldsymbol{u} \\
&= \int_{\mathcal{X}} \frac{f(\boldsymbol{x})}{\prod_{i=1}^d f_i\left(x_i\right)} \log\left(\frac{f(\boldsymbol{x})}{\prod_{i=1}^d f_i\left(x_i\right)}\right) \prod_{i=1}^d f_i\left(x_i\right)\, d\boldsymbol{x} \\
&= \int_{\mathcal{X}} f(\boldsymbol{x}) \log\left(\frac{f(\boldsymbol{x})}{\prod_{i=1}^d f_i\left(x_i\right)}\right) d\boldsymbol{x} \\
&= I\left(\boldsymbol{X}\right)
\end{aligned}$$

where the third equality follows from a change of variables with the trivial substitution $u_i = F_i(x_i)$ such that $du_i = f_i(x_i)\, dx_i$. This concludes the proof.
□

Finally, before moving on to correlation as a measure of similarity, we discuss what happens in the limit of mutual information and entropy as we shall later need this as arguments for numerical stability.

3.2.3 Entropy and mutual information in the limit

In this section, we shall discuss the differences between entropy and differential entropy and observe how this difference cancels when computing mutual information. In fact, we shall see that mutual information defined for continuous random vectors is the limit of the discrete version which will be useful later when implementing the algorithm.

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

$$\mathbb{P}\left(X^\Delta = x_i\right) = f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)\,dx \tag{3.7}$$

Clearly, this discretization is a valid distribution as

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

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

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

Clearly, the first term in the above expression converges to the differential entropy $h(X)$ as $\Delta \to 0$ whereas $\log \Delta \to -\infty$ i.e. the expression diverges altogether when differential entropy is well-defined.

A similar argument for the joint entropy between the discretization of X_1 and X_2 (and in principle to any number of dimensions), denoted by H_{12}^Δ, results in

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

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

$$f\left(x_1^{(i)}, x_2^{(j)}\right)\Delta_1\Delta_2 = \int_{j\Delta_2}^{(j+1)\Delta_2}\int_{i\Delta_1}^{(i+1)\Delta_1} f(x_1,x_2)\,dx_1dx_2, \quad \forall i,j \in \mathbb{Z}$$

Note that clearly $\left(x_1^{(i)}, x_2^{(j)}\right)$ exists for all $i, j \in \mathbb{Z}$. Again, the joint entropy diverges however, when computing the mutual information, we see that the diverging terms cancel. Namely, from Lemma 3.6

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

Thus, the limit of the mutual information for discrete random variables is indeed the mutual information defined for continuous random variables and can be computed either as the limit of discretizing the probability density function and then computing entropies or just using the initial definition for (discrete) mutual information in Definition 3.4.

Before continuing, we discuss the case where X_1 is equal to X_2. In this case, discretizing with a common Δ we have that

$$f\left(x_1^{(i)}, x_2^{(j)}\right)\Delta^2 = \int_{j\Delta}^{(j+1)\Delta}\int_{i\Delta}^{(i+1)\Delta} f(x_1,x_2)\,dx_1dx_2, \quad \forall i,j \in \mathbb{Z}$$

Clearly, the above integral is 0 for $i \neq j$. Although $f(x_1, x_2)$ is not well-defined in the usual functional sense, extending to distribution, we might write $f(x_1,x_2) = f(x_2|x_1)f(x_1)$. In terms of distributions, it works to put $f(x_2|x_1) = \delta(x_2 - x_1)$ where δ is the *Dirac delta* distribution, as then $\int_{\mathbb{R}} f(x_1,x_2)\,dx_2 = f(x_1)$ and $f(x_1,x_2)$ is "0" when $x_1 \neq x_2$. I.e. the right marginals and probability mass 1.

Then, when calculating the above integral, we get that

$$\begin{aligned} f\left(x_1^{(i)}, x_1^{(i)}\right)\Delta^2 &= \int_{i\Delta}^{(i+1)\Delta}\int_{i\Delta}^{(i+1)\Delta} f(x_1, x_2)\, dx_1 dx_2 \\ &= \int_{i\Delta}^{(i+1)\Delta} f\left(x_1\right)\, dx_1 \\ &= f\left(\tilde{x}_1^{(i)}\right)\Delta \end{aligned}$$

Thus, when calculating $I_{1,2}^{\Delta}$ we obtain

$$\begin{aligned} I_{1,2}^{\Delta} &= \sum_{i\in\mathbb{Z}} f_1\left(\tilde{x}_1^{(i)}\right)\log f_1\left(\tilde{x}_1^{(i)}\right)\Delta + \sum_{j\in\mathbb{Z}} f_2\left(\tilde{x}_2^{(j)}\right)\log f_2\left(\tilde{x}_2^{(j)}\right)\Delta \\ &\quad - \sum_{i\in\mathbb{Z}} f_1\left(\tilde{x}_1^{(i)}\right)\log f_1\left(\tilde{x}_1^{(i)}\right)\Delta - \log\Delta \\ &\to \infty \text{ as } \Delta \to 0 \end{aligned}$$

Thus in practice, it would not make much sense to compare equal variables or even a random vector only defined on a lower dimensional manifold as we would get an infinite Copula entropy.

3.2.4 Correlation

At this point, we have a good understanding of Copula entropy/mutual information for calculations later on. However, another typical measure of similarity is correlation which is easily estimated from sample data. However, in this section we show that in general, we can not infer the correlation coefficient from a copula. Namely, given a copula C for some set of random variables $\{X_i\}_{i\in I}$ indexed by finite I, one can not calculate ρ between any pair (X_i, X_j), $i \neq j$ from the copula. This is easily shown by the following argument.

First, note that from Corollary 3.2.1, C is also a copula for $Z_i := (X_i - \mu_i)/\sigma_i$ for $\in\in I$ where $\mu_i = \mathbb{E}[X_i]$ and $\sigma_i = \sqrt{\operatorname{Var} X_i}$. Clearly, the correlation coefficient for Z_i and Z_j is the same as between X_i and X_j. We thus proceed trying to calculate the correlation between any pair Z_i and Z_j.

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

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

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

again, under the assumption that Z_i has mean 0.

Thus, we conclude that indeed mutual information and correlation is very different measures of codependency (as correlation depends on the marginals whereas mutual information does not) and that it does not make much sense to introduce copulas in the setting of correlation albeit at this point we do not favor one measure above the other except if marginals should be insignificant to the network, Copula entropy is preferred.

3.3 Copula based network discovery

In this section, we will present the general algorithm and discuss some of its properties regarding uncertainty and convergence. We will focus on using mutual information i.e. Copula entropy as the measure of similarity but other measures such as correlation can be interchanged at will in the general algorithm.

By Theorem 3.8 we can compute the mutual information from observed data from the copula. Namely, let CE_{ij} denote the (pairwise) Copula entropy of variables X_i and X_j. We shall then set

$$G_{obs} = \begin{bmatrix} 0 & -CE_{12} & \dots & -CE_{1n} \\ -CE_{21} & 0 & \dots & -CE_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -CE_{n1} & -CE_{n2} & \dots & 0 \end{bmatrix} \tag{3.8}$$

where n is the number of nodes in the graph i.e. random variables that we have observed. Notice that we have chosen the diagonal elements as 0 since information between a random variable X and itself is not really well-defined and when trying to compute this numerically, we observe diverging results as also discussed in the previous section. Furthermore, only the information that propagates through the network is of interest and so setting 0 in the diagonal avoids a bias when (de)convolving the information or any similarity in general. Especially for mutual information where the information between a variable and itself diverges to ∞ thus in the limit, from Equation 3.5, we would get the identity matrix which does not tell us much about the direct dependencies.

Algorithm 1 then follows immediately from Equation 3.8

Algorithm 1 G_{obs} computation

```
Require: $n > 0$                                    ▷ Number of variables
  $G_{obs} \leftarrow \mathbf{0}$
  for $1 \leq i, j \leq n \mid i \neq j$ do
      Estimate $F_i$ and $F_j$ from $x_i^{\mathcal{D}}$ and $x_j^{\mathcal{D}}$
      $u_i^{\mathcal{D}} \leftarrow F_i(x_i^{\mathcal{D}})$
      $u_j^{\mathcal{D}} \leftarrow F_j(x_j^{\mathcal{D}})$
      Estimate $c_{ij}$ from $u_i^{\mathcal{D}}$ and $u_j^{\mathcal{D}}$
      Compute $NCE_{ij}$
      $[G_{obs}]_{ij} \leftarrow -NCE_{ij}$
  end for
  return $G_{obs}$
```

Namely, for each entry in G_{obs}, except for the diagonal elements, first estimate the cumulative distributions of X_i and X_j based on samples $x_i^{\mathcal{D}}$. Then, transform the samples by the estimated distribution function to obtain corresponding uniform samples. This may be done outside the loop to increase computational efficiency. From the paired samples $(x_i^{\mathcal{D}}, x_j^{\mathcal{D}})$, estimate the Copula density c_{ij} and finally use this to compute the mutual information/Copula entropy. Methods for estimating the densities and in continuation hereof the distribution functions are presented in **??**. The negative Copula entropy is then recorded in (i, j) entry of G_{obs}. We note that the algorithm can be optimized for symmetric measures such as Copula entropy itself, to only loop through $i < j$ and saving the computed entropy in the (j, i) entry as well. Also, as Copula entropy diverges as X_i and X_j are jointly distributed closer to a one-dimensional manifold, ideally there should be a check for such or the user should check the paired observations to exclude such variable combinations.

From Subsection 3.2.3, to calculate the (joint) copula entropy of a continuous random vector, we simply discretize the domain of each random variable and use the estimated copula density evaluated at these points to estimate the total Copula entropy. Furthermore, if one or more elements of the random vector are mixed random variables, we choose the discrete events to be their own bins and discretize the rest or in the context of Algorithm 1 only estimate the distribution functions for the continuous component of the random variable. This works due to the Copula entropy for continuous random variables being the limit of the discretization and as such, the Copula entropy is well-defined for mixed random variables as well.

We continue with an example of how this discretization of a mixed random variable would work. Notice that we only have a discrete event (an atom) at 0 as this resembles the observed behavior of the delays, although the example could be extended to more complex discrete distributions.

Example 3.1 (Discrization of mixed random variable). *Let X be a mixture of an atom in e.g. 0 and an exponential with parameter λ with proportions p and $1-p$. Then, a discretization of X is 0 with probability mass p and the remaining support $(0,\infty)$ discretized in some way with total probability mass $1-p$ and each bin having probability according to Equation 3.7 scaled with $1-p$. If the bin size is a constant Δ, then for the discretized variable X^Δ, we have $\mathbb{P}\left(X^\Delta = 0\right) = p$ and $\mathbb{P}\left(X^\Delta = x_i\right) = (1-p)\exp\left(-\lambda i\Delta\right)\left(1-\exp(-\lambda\Delta)\right)$, where x_i is given by*

$$x_i = i\Delta + \frac{1}{\lambda}\left(\log\left(\lambda\Delta\right) - \log\left(1-e^{-\lambda\Delta}\right)\right), \quad i \in \mathbb{N}_0$$

3.3.1 Network deconvolution

At this point, we have obtained a convolved matrix of information G_{obs} and are ready to use Equation 3.5. We present the original algorithm from [2] in the case G_{obs} is symmetric and hence diagonalizable by an orthogonal matrix U. The original `Matlab` implementation was translated to `Python` and is summarized in the following pseudocode.

Algorithm 2 (ND) Network Deconvolution

```
Require: $G_{obs}$, $\alpha$, $\beta$
    $[G_{obs}]_{ii} \leftarrow 0$, $\forall i \in \{1,\dots,N\}$          $\triangleright$ ensure zero-diagonal
    $[G_{obs}]_{ij} \leftarrow 0$, when $[G_{obs}]_{ij} < Q_\alpha\left(G_{obs}\right)$
    Compute eigendecomposition $U, \Lambda$ of $G_{obs}$
    $\lambda^+ \leftarrow \max\left(\lambda^{\max}, 0\right)$
    $\lambda^- \leftarrow \min\left(\lambda^{\min}, 0\right)$
    $c_s^+ \leftarrow \frac{1-\beta}{\beta}\lambda^+$
    $c_s^- \leftarrow \frac{1+\beta}{\beta}\lambda^-$
    $c_s \leftarrow \max\left(c_s^+, -c_s^-\right)$
    $\hat{\Lambda} \leftarrow \Lambda\left(c_s^{-1}I + \Lambda\right)^{-1}$
    return $U\hat{\Lambda}U^T$
```

where $Q_\alpha\left(G_{obs}\right)$ denotes the α quantile of the strictly upper (or lower due to symmetry) triangular part of G_{obs}. We note the two extra parameters *alpha* and β which we will discuss shortly. In particular, the paper contains conflicting information on how to find β from how it is defined. Furthermore, they include some analysis on the robustness of the above deconvolution algorithm but only in a somewhat particular case and with some confusion on matrix norms and spectral radius. This analysis on robustness, we will extend and clarify in the following Subsection 3.3.3.

From the definition of $Q_\alpha\left(G_{obs}\right)$ it is clear that the α parameter is a filter on the observed edges and is useful if one wants to filter out insignificant observations. However, in practice, as we will see, it is often not very influential except for large α (corresponding to many edges set to 0) as small perturbations from e.g. imperfect calculations should not influence the results for fairly conditioned matrices as we shall observe in Subsection 4.1.1. Thus, setting $\alpha = 0$ retains all values in G_{obs} after setting the diagonal equal to 0. As a technical detail, we note that the `quantile` function from `NumPy` (v. 1.26.4) has been used to find this quantile as quantiles can be defined in many ways from a data set.

Finally, we note that the $\beta \in (0,1)$ parameter corresponds to a scaling of G_{obs} such that the resulting spectral norm of G_{dir} is β. From Algorithm 2 it is seen that it serves as a regularization on the eigenvalues of G_{obs} and although this is discussed in [2], their results do not conform with their implementation, and we thus comment on this and what else could be done to ensure convergence of the algorithm in the following section. Also, in practice we choose a threshold t on the elements of G_{dir} returned from Algorithm 2 to further filter out insignificant direct dependencies.

3.3.2 Ensuring convergence and the effect of β

In this section we will further discuss the effect of β and how the steps for rescaling the observed similarity matrix G_{obs} are derived. In particular, we will reformulate the original derivation from [2] as there is a discrepancy between their code[1] and their proof of choosing a scaling parameter c_s of G_{obs}. Namely, denote $\tilde{G}_{obs}$ as the rescaled G_{obs} such that $\tilde{G}_{obs} = c_s G_{obs}$. Choosing c_s as in Algorithm 2 i.e. $c_s = \max\left(\frac{1-\beta}{\beta}\lambda^+, -\frac{1+\beta}{\beta}\lambda^-\right)$ where λ^+ is the largest positive eigenvalue of G_{obs} (and 0 if no eigenvalue is positive) and λ^- is the most negative eigenvalue of G_{obs} (and 0 if no eigenvalue is negative) then implies $\tilde{G}_{dir}$ obtained from the new $\tilde{G}_{obs}$ has spectral radius $\beta < 1$ i.e. a proper G_{dir} with the largest numerical eigenvalue equal to β. This holds in general and not only for symmetric G_{obs} as we will see in the following. However, when G_{obs} is symmetric the resulting $\tilde{G}_{dir}$ can easily be expressed through the eigendecomposition

[1]https://compbio.mit.edu/nd/

of G_{obs}, U, Λ as

$$\begin{aligned}
\tilde{G}_{dir} &= \tilde{G}_{obs}\left(I + \tilde{G}_{obs}\right)^{-1} \\
&= c_s G_{obs}\left(I + c_s G_{obs}\right)^{-1} \\
&= U c_s \Lambda U^T \left(UU^T + U c_s \Lambda U^T\right)^{-1} \\
&= U c_s \Lambda U^T U \left(I + c_s \Lambda\right)^{-1} U^T \\
&= U\Lambda \left(c_s^{-1} I + \Lambda\right)^{-1} U^T
\end{aligned}$$

which can also be seen in Algorithm 2. Thus, with everything else explained about the algorithm, we show that the resulting $\tilde{G}_{dir}$ in general have spectral radius β.

Let (λ, v) be an eigenpair of G_{obs} with $\lambda \neq 0$, it then follows that $\left(\frac{\lambda}{c_s^{-1}+\lambda}, v\right)$ is an eigenpair of $\tilde{G}_{dir}$. Then, following the arguments in [2] (which we have redone to know why the original implementation and derivation differs), we obtain that for a λ in $[0, \infty)$, $(-c_s^{-1}, 0)$ or $(-\infty, 0]$ to be mapped to $[-\beta, \beta]$ we have that

$$c_s^{-1} \geq \frac{1-\beta}{\beta}\lambda^+, \quad c_s^{-1} \geq -\frac{1+\beta}{\beta}\lambda^-$$

Thus, the smallest c_s^{-1} we can choose to ensure that $\rho\left(\tilde{G}_{dir}\right) \leq \beta$ is by $c_s = \min\left(\frac{1-\beta}{\beta}\lambda^+, -\frac{1+\beta}{\beta}\lambda^-\right)$ which also implies that $\rho\left(\tilde{G}_{dir}\right) = \beta$ as either the most negative or most positive eigenvalue is mapped to β or $-\beta$ respectively. This coincides with the original implementation, noting that some error has been made in the original discussion of the parameter β in [2]. Furthermore, we note that if we just want the algorithm to converge, as we discussed before, this is equivalent to $\sigma\left(\tilde{G}_{obs}\right) \subseteq (-1/2, \infty)$, so really, we can just choose $c_s^{-1} = -(2+\delta)\lambda^-$ for some small δ if $\lambda^- < -1/2$ and otherwise not scale G_{obs} to preserve the structure. Finally, we note that as β tends to 0, higher order interactions become less significant as can clearly be seen from Equation 3.4. Thus, β also allows us to tune how much influence higher order interactions should have and one should try different β to see how influenced results are to higher order effects.

3.3.3 Robustness to noise

Finally, before discussing how to compute and estimate the mutual information between two random variables based on observations, we turn our heads to error analysis of the deconvolution algorithm. It is important to understand how well the algorithm performs subject to noise and errors. Namely, in the case of mutual information, the assumption that higher order effects can be calculated as a sum of matrix powers of the direct effects does not hold. Thus, if we can say quantize the error in G_{obs}, we can from the following analysis quantize the resulting error in G_{dir}. We shall first discuss the original result from [2], correcting some errors in terms of definitions and see how their result can also be expressed as an absolute upper bound on the error instead of only how this error behaves for small perturbations. Furthermore, we shall extend their result to not only hold when $\rho\left(G_{obs}\right) < 1$ and $\rho\left(G_{obs} + N\right) < 1$ where N is some noise e.g. from computation or assumptions that does not completely hold.

The original result states that $\left|\left|G_{dir} - \tilde{G}_{dir}\right|\right|_2 \leq \gamma + \mathcal{O}\left(\delta^2 + \gamma^2 + \delta\gamma\right)$ where $||\cdot||_2$ is the Euclidean norm also known as the spectral norm as this is equal to the largest singular value of the input matrix. However, they note that the Euclidean norm of a matrix M is equal to $\sqrt{\sum_{i,j} m_{ij}^2}$ which is incorrect. This is the Frobenius norm, and instead it should have been defined as

$$||M||_2 = \sup_{||x||_2=1} ||Mx||_2 = \sigma_{\max}\left(M\right)$$

They then proceed to let γ be the largest absolute eigenvalue of N and δ the largest absolute eigenvalue of $\tilde{G}_{obs} = G_{obs} + N$ however as the noise may be both positive and negative, it is easier to define δ as the largest absolute eigenvalue of G_{obs} instead which we will do in the following. We note that γ and δ are not the spectral/Euclidian norm of N and G_{obs} respectively as in general, we only have $\rho\left(M\right) \leq ||M||_2$. However, if G_{obs} and N are both (real) symmetric matrices, then the spectral norms are equal to the largest absolute eigenvalues of G_{obs} and N respectively. Thus, if instead one wanted to measure the difference in the direct dependency matrices in terms of e.g. the Frobenius norm, it is important to differentiate between the spectral radius and the norm that is actually being used. Finally, before constructing the actual upper bound on the error instead of quantizing the asymptotic behavior for small γ, we note that $||\cdot||_2$ is a sub-multiplicative matrix norm defined as bellow ([6]), and that we shall assume that $\rho\left(G_{obs}\right), \rho\left(\tilde{G}_{obs}\right) < 1$.

Definition 3.9 (Sub-multiplicative Matrix norm)**.** *A matrix norm* $|||\cdot|||$ *is said to be sub-multiplicative, if for every* $A, B \in \mathbb{F}^{n\times n}$ *where* $\mathbb{F}$ *is either the real or complex field:*

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

As we do not use any property of the spectral norm except that it is sub-multiplicative, we shall consider any norm $||\cdot||$ in general that is also sub-multiplicative. Thus, consider the norm of the difference $G_{dir} - \tilde{G}_{dir}$:

$$\begin{aligned}
\left|\left|G_{dir} - \hat{G}_{dir}\right|\right| &= \left|\left|G_{obs}\left(I + G_{obs}\right)^{-1} - \hat{G}_{obs}\left(I + \hat{G}_{obs}\right)^{-1}\right|\right| \\
&= \left|\left|-\sum_{k\geq 1}\left(-G_{obs}\right)^k + \sum_{k\geq 1}\left(-\hat{G}_{obs}\right)^k\right|\right| \\
&\leq \sum_{k\geq 1}\left|\left|G_{obs}^k - \left(G_{obs} + N\right)^k\right|\right| \\
&\leq \sum_{k\geq 1}\sum_{i=1}^{k}\binom{k}{i}\left|\left|N\right|\right|^i\left|\left|G_{obs}\right|\right|^{k-i} \\
&= \sum_{k\geq 1}\sum_{i=1}^{k}\binom{k}{i}\gamma^i\delta^{k-i} \\
&= \sum_{k\geq 1}\left(\left(\gamma+\delta\right)^k - \delta^k\right) \\
&= \frac{\gamma+\delta}{1-\gamma-\delta} - \frac{\delta}{1-\delta} \\
&= \frac{\gamma}{(1-\gamma-\delta)(1-\delta)}
\end{aligned} \tag{3.9}$$

where in the second to last inequality, we assume that $\gamma + \delta < 1$ as then both $\sum\left(\gamma+\delta\right)^k$ and $\sum\delta^k$ converges as $\gamma + \delta \geq \delta \geq 0$ and hence also the difference of the sums converges. Also, the second equality uses that the spectral norm of G_{obs} and $\tilde{G}_{obs}$ is less than 1 in order to express the inverses as infinite series. Thus, the above bound on the difference $G_{dir} - \tilde{G}_{dir}$ does not hold in every case and we observe for fixed γ, the bound tends to ∞ as $\delta \to 1$. Furthermore, we note that the final infinite sum diverges whenever $\gamma+\delta > 1$ through the following argument using the ratio test for infinite sums which is needed because we can not conclude on the convergence of a difference of diverging sums solely from the fact that the individual sums diverge:

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

assuming that $\gamma, \delta > 0$ corresponding to neither N nor G_{obs} is the zero matrix in which case the above analysis in nonsensical.

Before continuing with a more general bound on the error, we first note that examples of sub-multiplicative matrix norms are every induced norm such as the spectral norm and the Frobenius norm which is often useful when interpreting error. Also, the max norm is *not* sub-multiplicative, but a scaled version is (which is true for any matrix norm from the fact that all matrix norms are equivalent).

Now, consider the general case, where we do not restrict the spectral radius of either G_{obs} or N except such that G_{obs} and $\tilde{G}_{obs}$ admits direct similarity matrices G_{dir} and $\tilde{G}_{dir}$ (with spectral radius less than 1). To obtain a more general result, we shall use the following result from [7], which is very useful when doing matrix perturbation analysis.

Theorem 3.10 (Inverse of sum of matrices)**.** *Let $A, B \in \mathbb{R}^{n\times n}$ such that A and $A+B$ are invertible. Then the inverse of $A+B$ can be expressed as*

$$(A+B)^{-1} = A^{-1} - A^{-1}B\left(A+B\right)^{-1}$$

The proof of the above is simple through direct computation. Hence, we continue to once again consider the difference $G_{dir} - \tilde{G}_{dir}$

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

where the third equality follows from Theorem 3.10. This way, we have a simple exact expression for the difference without any further assumptions on G_{obs} and N. Now, under a sub-multiplicative norm $||\cdot||$ we can bound the norm of the difference in the following way.

$$\left|\left|G_{dir} - \tilde{G}_{dir}\right|\right| \leq ||N|| \left|\left|\left(I + G_{obs}\right)^{-1}\right|\right| \left|\left|\left(I + G_{obs} + N\right)^{-1}\right|\right| \tag{3.10}$$

We note that if once again, we assume that the spectral radius of G_{obs} and $G_{obs} + N$ are smaller than 1, we rediscover Equation 3.9. Equation 3.10 also shows that in general, if N is small or G_{obs} is large we should observe small errors which is also what we would expect intuitively. The above result is also very useful when later on in Subsection 4.1.1 we discuss the error from using mutual information in the case of a multi-variate Gaussian.

From Equation 3.10, by another application of Theorem 3.10, we find the relative error in general to be bounded as follows

$$\frac{\left|\left|G_{dir} - \tilde{G}_{dir}\right|\right|}{||G_{dir}||} \leq ||N|| \left|1 - \frac{||I||}{\left|\left|G_{obs}\left(I + G_{obs}\right)^{-1}\right|\right|}\right| \left|\left|\left(I + G_{obs} + N\right)^{-1}\right|\right|$$

Finally, before discussing the methods for estimating the copula density, we comment on some frequently used matrix norms and show some explicit bounds on the error only using the difference of G_{obs} and $\tilde{G}_{obs}$, N. Namely, we shall consider the max norm and Frobenius norm of the difference $G_{dir} - \tilde{G}_{dir}$ and note that from [8], we can relate the Euclidean norm to the Frobenius and max norm in the following way. Namely, for any matrix $A \in \mathbb{R}^{n \times n}$ it holds that

$$||A||_2 \leq ||A||_F \leq \sqrt{n}\,||A||_2$$

$$||A||_{\max} \leq ||A||_2 \leq n\,||A||_{\max}$$

Finally, if G_{obs} and N are symmetric, the singular values are equal to the absolute eigenvalues for G_{obs} and $\tilde{G}_{obs}$ and because $\sigma\left(I + G_{obs}\right), \sigma\left(I + \tilde{G}_{obs}\right) \subseteq (1/2, \infty)$ implies $\sigma\left(\left(I + G_{obs}\right)^{-1}\right), \sigma\left(\left(I + \tilde{G}_{obs}\right)^{-1}\right) \subseteq (0, 2)$ we infer that $\left|\left|\left(I + G_{obs}\right)^{-1}\right|\right|_2, \left|\left|\left(I + \tilde{G}_{obs}\right)^{-1}\right|\right|_2 \leq 2$. Using this with the above equivalence on the Euclidean norm with Equation 3.10, we conclude that

$$\begin{aligned} \left|\left|G_{dir} - \tilde{G}_{dir}\right|\right|_F &\leq 4\sqrt{d}\,||N||_F \\ \left|\left|G_{dir} - \tilde{G}_{dir}\right|\right|_{\max} &\leq 4d\,||N||_{\max} \end{aligned} \tag{3.11}$$

This clearly shows us that for small networks (thus small d) we risk smaller errors in terms of the Frobenius and max norm (which is not surprising) which are clearly interpreted through the difference of individual element of G_{dir} and $\tilde{G}_{dir}$ and that the max norm scales linearly with the number of nodes while the Frobenius difference only scales with the square root of the number of nodes.

3.3.4 KDE methods

Det her mangler lige

CHAPTER 4

Results

4.1 Examples

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

4.1.1 Gaussian CE error

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

Proposition 4.1. *Given a bivariate normal distribution* $\boldsymbol{X} \sim \mathcal{N}\left(\boldsymbol{\mu}, \Sigma\right)$ *where*

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

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

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

Example 4.1. *Exponentiated multivariate Gaussian*

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

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

It is clear that to Algorithm 1, the mean is of no importance as it simply corresponds to a scaling of the Y_i *variables. Furthermore, because of Corollary 3.2.1, theoretically, due to the uniqueness of the Copula* C *(as* $\boldsymbol{Y}$ *is continuous) we should expect near equal or very similar results for* $\boldsymbol{Y}$ *and* $\boldsymbol{X}$ *from Algorithm 1. Additionally, different* σ *corresponds to different scaling of* $\boldsymbol{X}$*, and thus we should observe equal or near equal* G_{dir} *for all* $\boldsymbol{Y}$*. Initially, we shall see how this hypothesis holds up to the following three examples*

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

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

$$G_{dir} = \begin{bmatrix} -0.33396 & 0.6660 & 0.02512 \\ 0.6660 & -0.3341 & 0.02730 \\ 0.02512 & 0.02730 & -0.0020583 \end{bmatrix} \tag{4.1}$$

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

$$G_{dir} = \begin{bmatrix} -0.3335 & 0.6665 & 0.01414 \\ 0.6665 & -0.3335 & 0.01418 \\ 0.01414 & 0.01418 & -0.00060124 \end{bmatrix} \tag{4.2}$$

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

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

For $\boldsymbol{\sigma} = (1, 1, 1)$ *and* $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ *we observe the most resemblance to the* Σ*, although the resulting* G_{dir} *deviate in the final column. The difference is likely produced by Algorithm 1 as if the resulting* G_{obs} *was the same, then so would* G_{dir} *and from the above argument, we know that theoretically this should be the case. For the final example,* $\boldsymbol{\sigma} = (1, 2, 3)$*, we see a completely different result and immediately suspect that there must be some numerical errors. Investigating* *the partial results of Algorithm 1 we immediately see a flaw in the supposedly uniform variables U_i as shown in figure Figure 4.1*

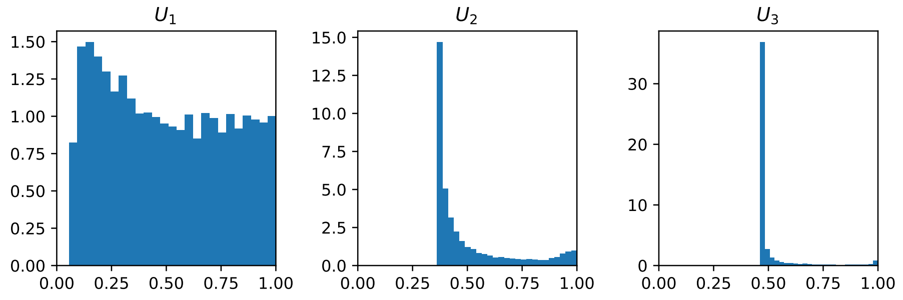


Figure 4.1: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 2, 3)$. These should be uniformly distributed, but clearly this is not the case for U_2 and U_3. Even U_1 does not quite resemble 10,000 samples from a uniform distribution.

	U_1	U_2	U_3
D_n	0.066209	0.36014	0.46285
p-value	0	0	0

Table 4.1: based on 10,000 samples for $\boldsymbol{\sigma} = (1, 2, 3)$.

Before handling this, the non-uniformity of U_1 in Figure 4.1 is likely also present in the case when $\boldsymbol{\sigma} = (1, 1, 1)$. Indeed, Figure 4.2 shows that this is indeed the case.

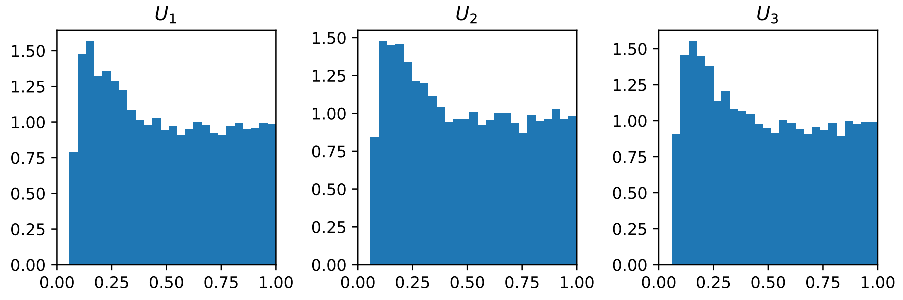


Figure 4.2: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (1, 1, 1)$.

Finally, just to be sure, $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$ is also shown in Figure 4.3 and seems very reasonable, except for U_3.

	U_1	U_2	U_3
D_n	0.068408	0.066808	0.070809
p-value	0	0	0

Table 4.2: based on 10,000 samples for $\boldsymbol{\sigma} = (1, 1, 1)$.

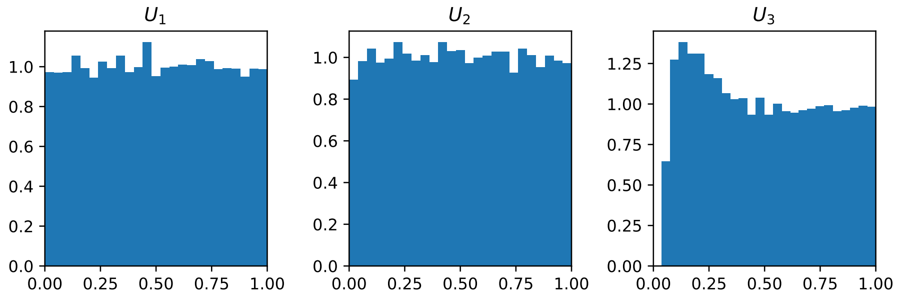


Figure 4.3: The samples transformed using $U_i = F_i(X_i)$ for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

	U_1	U_2	U_3
D_n	0.00581897	0.0068066	0.050908
p-value	0.88645	0.74179	0

Table 4.3: based on 10,000 samples for $\boldsymbol{\sigma} = (0.07, 0.3, 0.9)$.

From the above examples, it seems that the larger the variance, the worse the uniforms turn out. Reasons for this could include numerical issues when trying to calculate $u_i^{(j)}$ form $y_i^{(j)}$ by $u_i^{(j)} = \int_{-\infty}^{y_i^{(j)}} f_i(y)\,dy$ and bad fitting of the kernel density estimate from observations. In particular, for values similar, which happens in the case for large σ such that we observe large negative realizations of X_i, $y_i^{(j)}$ are almost 0, and when computing the integral could result in identical values. Furthermore, from Figure 4.4 we see that indeed the fit is quite poor. Note that we have zoomed in on the interval $[-200, 200]$ which contains 96.2% of observations. The poor fit is primarily due to the use of Scott's Rule as discussed above which in this case overshoots the optimal bandwidth by a lot.

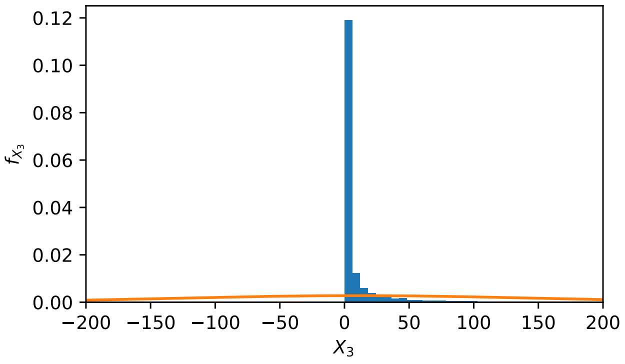


Figure 4.4

The poor fit also explains the high concentration of U_3 around 0.5 in Figure 4.1 as only 54.5% of the probability mass lies above 0.

However, also here Corollary 3.2.1 proves to be useful. Namely, we can get rid of the numerical issues by transforming Y_i using e.g. $\log(\cdot)$ or $(\cdot)^p$ for $p > 0$ to get even out the observations more. As the first simply inverts the initial transformation of X_i, we choose the latter as a more interesting case. In particular, choosing $p < 1$ will result in a more even distribution. In the following, $p = 1/10$ has been used to transform $\boldsymbol{Y}$ prior to running Algorithm 1 and the resulting $u_i^{(j)}$ is shown in Figure 4.5.

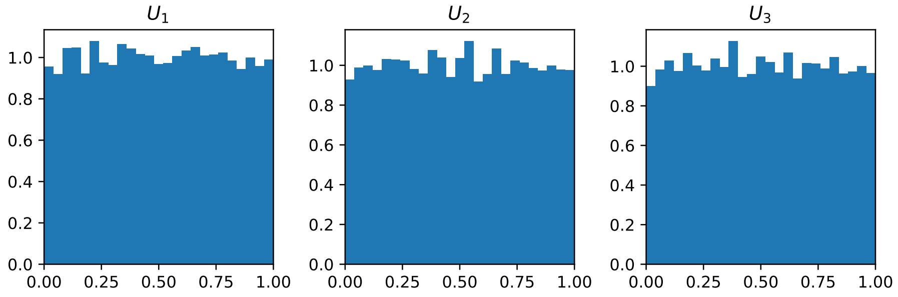


Figure 4.5

The resulting $u_i^{(j)}$ now seem to follow a uniform distribution and indeed the KDE fits much better as seen in Figure 4.6.

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

Table 4.4: based on 10,000 samples for $\boldsymbol{\sigma} = (1, 2, 3)$ with power transform.

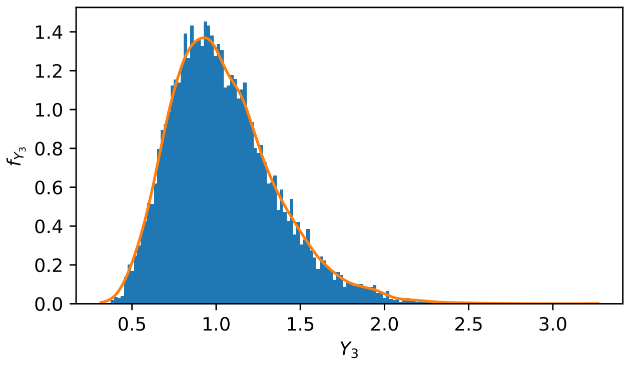


Figure 4.6

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

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

Which is indeed much more comparable with the result from before in Equation 4.1 and Equation 4.2. The difference between G_{dir} from $\mathbf{Y}$ and $\mathbf{Y}^p$ is clearly visible in Figure 4.7 and also Figure 4.7b resembles the original correlation structure.

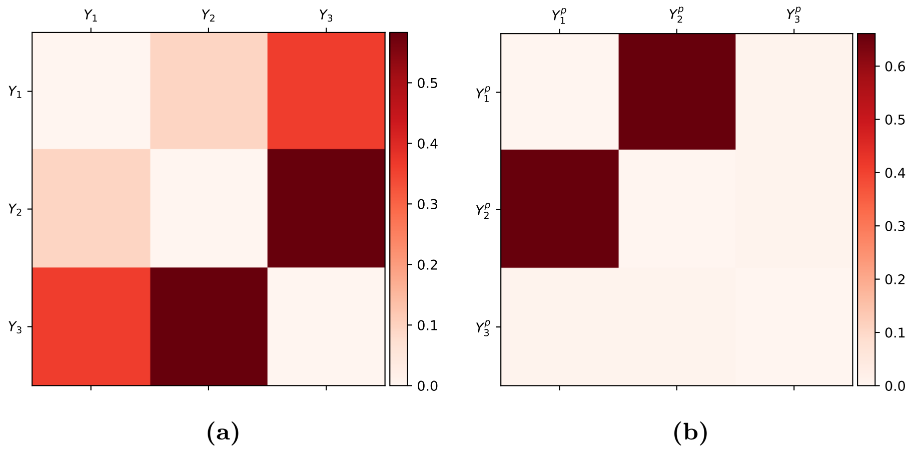


Figure 4.7: G_{dir} resulting from 10,000 samples from multi variate Gaussian with $\boldsymbol{\sigma} = (1, 2, 3)$ in **(a)** with raw samples from $\boldsymbol{Y}$ and in **(b)** the transformed data corresponding to $\boldsymbol{Y}^p$.

Finally, to end this example we shall compare with some theoretical results. Namely, the output G_{obs} of Algorithm 1 can also be calculated theoretically. For this, we shall use Proposition 4.1 which permits a theoretical result, namely

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

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

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

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

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

r is given by

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

from the mutual information, we can calculate the absolute correlation. Notice that the density does not change when reversing both r and ρ simultaniously, thus, wlog, assume $r \geq 0$, then we can calculate a CI for ρ (which will be negated if we had used $-r$ instead and thus would be identical when taking the absolute value). If the original CI $[a, b]$ contains 0 i.e. $a < 0$, we shall write the CI for the absolute correlation as $[0, b]$ instead. This way, we can compare the absoulte correlations and see if they are the same (by checking if the CI contains the theoretical correlation) by [9]. Using numerical integration (fast enough with high numerical accuracy from many bins, 1 mil bins, yielding probability mass 1.0000000000008133), can compute CI for absolute correlation

Section A.1

Clearly these are not equal, but in this case, the error is suspected to originate from the estimated joint density. For example, considering X_1 and X_2, we compare the estimated joint copula density and compare to the theoretical reference til et sted hvor gausisk copula står shown in Figure 4.8 and Figure 4.9 respectively.

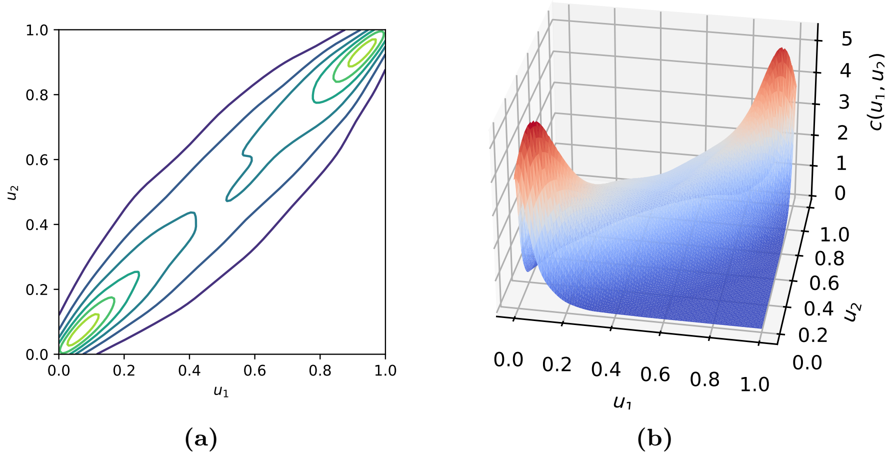


Figure 4.8: Estimated copula density c with $\rho = 0.9$ corresponding to X_1 and X_2.

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

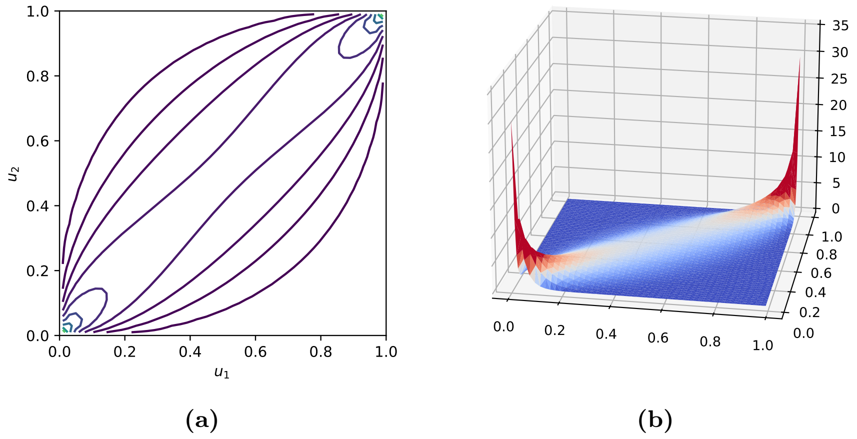


Figure 4.9: Theoretical copula density c with $\rho = 0.9$ corresponding to X_1 and X_2.

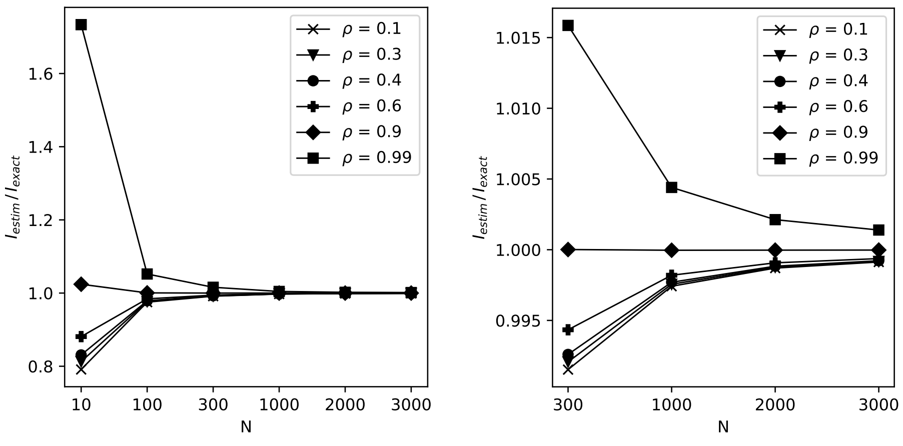
We note however, that the underlying structure is still captured i.e. that Y_1 and Y_2 covary while Y_3 does not inform Y_1 or Y_2 and vice versa.

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

Example 4.2. *From Example 4.1 we saw how one could handle some numerical issues. Thus, in this example we shall not bother ourselves with such computations and merely focus on the correlation structure. In particular, we shall sample $\boldsymbol{X}$ from a 10 dimensional*

4.2 sammenligning af metoder for at finde MI

Sammenligning af gammel metode og "min"



(a) (b)

Figure 4.10: Evaluation of MI for new method for different N. Bør sammenlignes med artiekl fundet (har sat i bibtex) og original papers (ikke Kina)

ved høj korrelation i.e. tæt på laver dimensionel manifold, skal der bruges mange, som i rigtig mange samples i mesh.

Inkluder flere exempler end blot gaussian as done by [10]

4.2.1 10D gaussian example

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

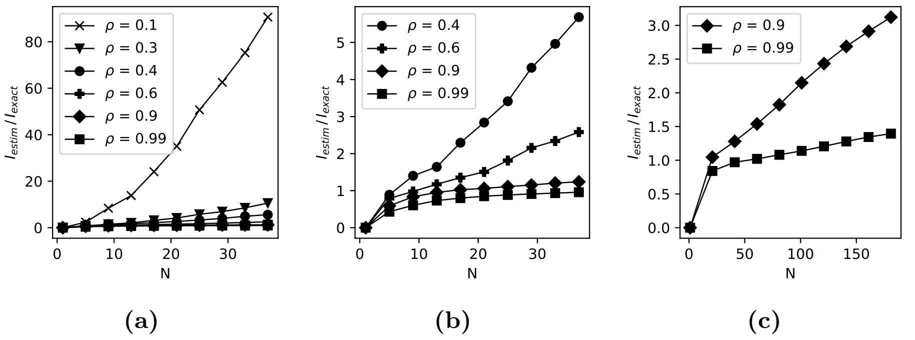


Figure 4.11: Evaluation of MI for old method for different N. Ligner der er knæk ved fporhold lig 1. Men ved næremere undersøgelse blev det fundet ud af at det ikke helt er tilfældet, og derudover vil der skulle laves en algoritmisk måde at finde dette knæk på. Savitzky–Golay filter kunne være en mulighed, eller gruppere e.g. 5 forskellige bins og tag gennemsnit. Efter smoothing kan anden afledte tæt på 0 bruges, til at finde hvornår stykket bliver fladt (tilnærmelses vist)

Appendix A

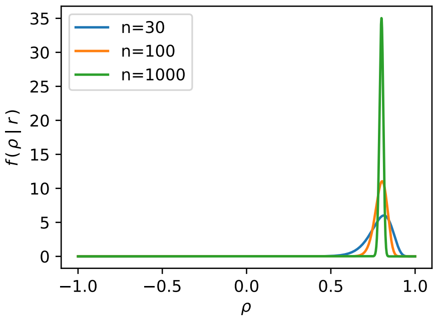
Appendix

A.1 Confidence interval for absolute correlation in bivariate Gaussian

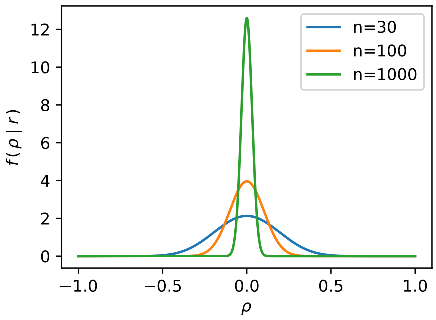
From [9], given a bivariate Gaussian, the confidence distribution of ρ given the empirical correlation r based on n observations is given by

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

where $F(a, b, c, z)$ is the Gaussian hypergeometric function and $\nu = n-1$. That is, given a sample correlation r, what is the confidence in ρ in terms of a distribution. In the following figure, a sample correlation $r = 0.8$ and $r = 0$ has been used with varying number of observations (degrees of freedom) in figures Figure A.1a and Figure A.1b respectively. A key property is that f is *even symmetric* in ρ, r. That is $f(\rho \mid r) = f(-\rho \mid -r)$. Thus, a confidence interval for ρ given r is the negative of the confidence interval given $-r$. In particular, if we only observe $|r|$, we can calculate a confidence interval for ρ up to the sign of the bounds of the interval. Furthermore, as we want a CI for $|\rho|$, it does not matter if r is negative or positive. Hence, without loss of generality, we assume that $r \geq 0$. At this point, to construct a confidence interval for $|\rho|$ we



(a)



(b)

Figure A.1: $f(\rho \mid r, \nu)$ shown for $r = 0.8$ and $r = 0$ in (a) and (b) with $n \in \{30, 100, 1000\}$. As one would expect, the power i.e. the width of the peak decreases with increasing n and for correlations closer to 0, the width is the largest.

list the following desired properties. Firstly, it should be an exact confidence interval, meaning that for a given significance level α, the CI includes the true value exactly $1-\alpha$ fraction of the times. Secondly, if for a given r, it can not be rejected that ρ is 0, 0 should also be contained in the interval. Finally, if we reject that $\rho = 0$, we shall have $\alpha/2$ probability mass above and below the bounds of the interval. The above is enough to uniquely define a confidence interval in all cases. Before continuing with how this CI is calculated, we mention that as r is an unbiased estimator of ρ, we would preferably want $|r| \in CI_{1-\alpha}(|\rho|)$ (where $CI_{1-\alpha}(|\rho|)$ denotes the $1-\alpha$ confidence interval for $|\rho|$). However, although this will in almost every scenario be the case, we can not be sure of this from the above properties and in fact examples with large α can be constructed such that $|r|$ lies just outside the constructed CI.

First, to conform with the second desired property, if it can not be rejected that $\rho = 0$ on a significance level α, we will initially compute a CI for ρ (not $|\rho|$) based on r (wlog chosen to be non-negative). This CI will just be a symmetric CI in the sense that $\alpha/2$ of the probability mass will lie below the lower bound of the CI and above the upper bound of the CI respectively. If 0 is contained in this CI, we can not reject that $\rho = 0$ and vice versa on an α significance level. Thus, if 0 is contained in this initial CI for ρ, we will start the CI for $|\rho|$ at 0 and determine and upper bound b such that α probability mass is above this b. Otherwise, we shall find a and b such that $\alpha/2$ probability mass is below a and above b respectively. Choosing a and b this way also conforms with the third property. Finally, to ensure that the CI contains exactly $1-\alpha$, we define $\tilde{f}$ as the reflected f in ρ such that

$$\tilde{f}\,(\rho_a \mid r_a, \nu) = f(\rho_a \mid r_a, \nu) + f(-\rho_a \mid r_a, \nu), \quad \rho_a, r \in [0, 1]$$

where ρ_a and r_a is the absolute correlation and empirical correlation respectively. With this $\tilde{f}$, the density at ρ_a is both the density for the negative and positive correlation ensuring that the $\tilde{f}$ has probability mass 1. Thus, if $a = 0$ (i.e. the CI must contain 0), we find b as the $1 - \alpha$ percentile of $\tilde{f}$ and if $a \neq 0$, we take a as the $\alpha/2$ percentile and b as the $1 - \alpha/2$ percentile of $\tilde{f}$.

As an example, suppose $r_a = 0.06$ with 1000 observations. Then a 95% CI for $|\rho|$ is $[0, 0.11164]$ whereas if on had observed $r_a = 0.07$ the CI would be $[0.01071, 0.1314]$. These CI could then be used to test the absolute correlation of a bivariate Gaussian i.e. for $r_a = 0.07$ based on 1000 observations would be rejected as stemming from a Gaussian with absolute correlation 0.01 on a 5% significance level.

Bibliography

[1] M. L. C. Vicente, "A benchmark model to generate batch process data for machine learning testing and comparison," 2021.

[2] S. Feizi, D. Marbach, M. Médard, and M. Kellis, "Network deconvolution as a general method to distinguish direct dependencies in networks," *Nature biotechnology*, vol. 31, pp. 726 – 733, 2013.

[3] D. Kurowicka and H. Joe, *Dependence Modeling: Vine Copula Handbook.* 12 2010.

[4] G. Geenens, "Copula modeling for discrete random vectors," *Dependence Modeling*, vol. 8, pp. 417–440, 12 2020.

[5] Y.-N. Sun, W. Qin, and Z. Zhuang, "Nonparametric-copula-entropy and network deconvolution method for causal discovery in complex manufacturing systems," *Journal of Intelligent Manufacturing*, vol. 33, 08 2022.

[6] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University Press, 2 ed., 2012.

[7] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *SIAM Review*, vol. 23, no. 1, pp. 53–60, 1981.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations.* The Johns Hopkins University Press, third ed., 1996.

[9] G. Taraldsen, "Confidence in correlation," 11 2020.

[10] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004.