

General introduction to metodeafsnittet og causalitet

1.1 Causal discovery

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

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

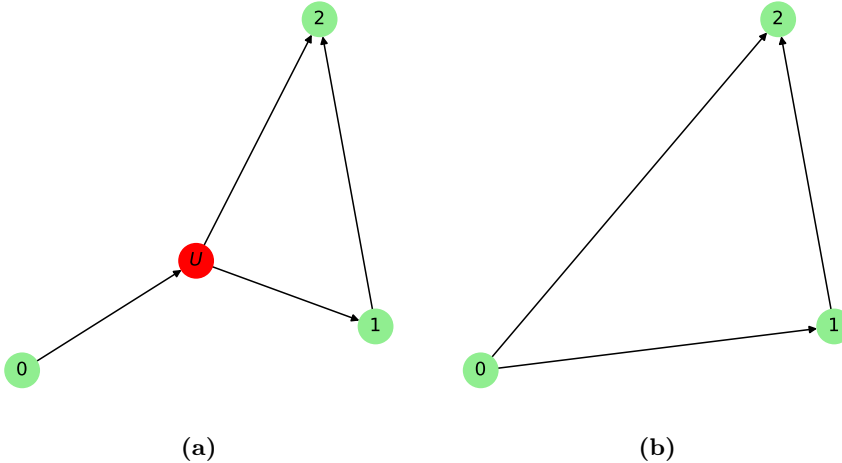


Figure 1.1: (a) An example of a causal structure depicted as a graph. When observing the network, only nodes 0, 1 and 2 are observed/recorded. (b) The resulting inferred graph from observational data. Although this is not a complete picture of the true underlying dynamics of the system, if only the observed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

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

1.1.1 Setup and assumptions

Suppose a set of N random variables (X_i) is given. The method presented in this section aims to discover direct relationships between pairs X_i and X_j for $i \neq j$. These relationships will be presented by a directed graph as in the previous section or an undirected graph in case the causal direction is either unknown or such an assumption on direction is not plausible. In particular, we shall let each random variable X_i be represented by a vertex in a graph. We

will later discuss a way of directing edges such that a causal network may be discovered i.e. a directed acyclic graph that may be used for inference.

The method proposed by [?] then works as follows. Given an observed matrix $G_{obs} \in \mathbb{R}^{N \times N}$ of similarities between each pair of variables, we shall deduce a matrix $G_{dir} \in \mathbb{R}^{N \times N}$ of direct similarities between each pair of random variables X_i and X_j . In particular, we wish to filter out indirect effects which we will denote by G_{dir} defined as effects between pairs of variables that is the result of effects propagating through other variables. 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 1.2 for a further discussion on these two measures and Section 1.3 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*.

illustration af hvordan algoritmen fungerer/formål

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 1.3.3, the algorithm is somewhat robust to noise in the sense that we can ensure accuracy depending on the level of noise observed present in G_{obs} and on the norm chosen (from a certain, although rather general, set of norms). Namely, if G_{obs} is subject to noise, we find a bound on how different the inferred directed effects can be to the true direct effects using different matrix norms to measure this difference. This hints to that a threshold on the inferred weights on the edges of the network might be a good idea to remove small inferred effects. This 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

$$G_{obs} = G_{dir} + G_{indir} \quad (1.1)$$

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

The second and final assumption is that the indirect effects G_{indir} can be com-

puted in terms of G_{dir} . Namely, that

$$G_{indir} = G_{dir}^2 + G_{dir}^3 + \dots \quad (1.2)$$

i.e. that the observed *information* exchanged on an edge e_{ij} between nodes X_i and X_j is the sum of the 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

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

where e_{ij} is the (i, j) element of G_{dir} . This is of course not true in general, but through the error analysis in Subsection 1.3.3 and the examples in chapter 1 this is not necessarily a problem. 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

$$G_{obs} = G_{dir} + G_{dir}^2 + G_{dir}^3 + \dots = G_{dir} + G_{dir} G_{obs} \quad (1.3)$$

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. $\rho(G_{dir}) < 1$. Thus, assuming convergence we can rewrite the infinite series as

$$G_{obs} = G_{dir} (I - G_{dir})^{-1} \quad (1.4)$$

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

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

Furthermore, if the measure of dependence between pairs of variables is symmetric, then so is G_{obs} and hence diagonalizable by some orthogonal matrix U 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 $\Lambda_{dir} = \Lambda_{obs} (I + \Lambda_{obs})^{-1}$ which is also a diagonal matrix.

We note that from the above one needs $(I + G_{obs})^{-1}$ 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 $\rho(G_{dir}) < 1$, and that $I + G_{obs}$ is thus invertible we use Equation 1.4 and simplify

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

which is clearly invertible. Furthermore, we note that under the assumption $\rho(G_{dir}) < 1$ 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 1.5, 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 1.8. 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, $T = D(I + D^{-1}T_u)$ and hence, we have that

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

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*.

1.2 Information measures and computation

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

1.2.1 Copula

Given a set of N random variables X_1, \dots, 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, \dots, F_N we define a copula C as

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

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

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

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

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

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) \quad (1.6)$$

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

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

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

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

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

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

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

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

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

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

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

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

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

1.2.2 Mutual information and Copula entropy

In this section we introduce Copula entropy as done in [?] and see how it actually is equal to the well known mutual information (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 1.4 (Mutual information). *For a random vector $\mathbf{X} = \{X_i\}$, we define the mutual information as*

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

where f is the joint density function with marginals f_i of the random vector \mathbf{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 1.2.3. For now, we proceed with the definition of (joint) entropy for both discrete and continuous random vectors.

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

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

In case of a discrete random vector, this is called the Shannon entropy while for continuous random vectors, this is called differential entropy and is often denoted as $h(\mathbf{X})$ instead of $H(\mathbf{X})$.

We note the need for two separate notations of entropy as differential entropy is not the limit of Shannon entropy in the way mutual information is. Again, this is further discussed in Subsection 1.2.3.

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

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

where d is the dimension of \mathbf{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:

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

□

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

Definition 1.7 (Copula entropy). *For a continuous random vector \mathbf{X} with a uniquely defined Copula C , and Copula density c , we define the Copula entropy CE of \mathbf{X} as*

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

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

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

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

Theorem 1.8 (Equality of Copula entropy). *For a continuous random vector \mathbf{X} , the Copula entropy CE is equal to the negative joint mutual information of \mathbf{X}*

$$CE(\mathbf{X}) = -I(\mathbf{X})$$

Proof. By Theorem 1.2, letting $x_i = F_i^{-1}(u_i)$, we can relate the copula density to the joint density of \mathbf{X} and its marginals

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

It follows directly that

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

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. \square

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.

1.2.3 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 :

$$\mathbb{P}(X^\Delta = x_i) = f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x) \, dx \quad (1.7)$$

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 $\Delta \rightarrow 0$ 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 $\Delta \rightarrow 0$ whereas $\log \Delta \rightarrow -\infty$ 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

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

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

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

Note that clearly $f(x_1^{(i)}, x_2^{(j)})$ 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 1.6

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

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 1.4.

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

$$f\left(x_1^{(i)}, x_2^{(j)}\right) \Delta^2 = \int_{j\Delta}^{(j+1)\Delta} \int_{i\Delta}^{(i+1)\Delta} f(x_1, x_2) dx_1 dx_2, \quad \forall i, j \in \mathbb{Z}$$

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

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

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

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

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.

1.2.4 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 compute the correlation coefficient from a Copula which we saw above is the case for mutual information. 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 1.2.1, C is also a copula for $Z_i := (X_i - \mu_i) / \sigma_i$ for $i \in I$ where $\mu_i = \mathbb{E}[X_i]$ and $\sigma_i = \sqrt{\text{Var } X_i}$. 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.

1.3 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 1.8 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

$$G_{obs} = \begin{bmatrix} 0 & -CE_{12} & \dots & -CE_{1n} \\ -CE_{21} & 0 & \dots & -CE_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -CE_{n1} & -CE_{n2} & \dots & 0 \end{bmatrix} \quad (1.8)$$

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 1.5, we would get the identity matrix which does not tell us much about the direct dependencies.

Algorithm 1 then follows immediately from Equation 1.8

Algorithm 1 G_{obs} computation

Require: $n > 0$ \triangleright 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 1.2.3, 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 1.1 (Discretization 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 1.7 scaled with $1-p$. If the bin size is a constant Δ , then for the discretized variable X^Δ , we have $\mathbb{P}(X^\Delta = 0) = p$ and $\mathbb{P}(X^\Delta = x_i) = (1-p) \exp(-\lambda i \Delta) (1 - \exp(-\lambda \Delta))$, where x_i is given by*

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

1.3.1 Network deconvolution

At this point, we have obtained a convolved matrix of information G_{obs} and are ready to use Equation 1.5. We present the original algorithm from [?] 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}, α, β

$[G_{obs}]_{ii} \leftarrow 0, \forall i \in \{1, \dots, N\}$ \triangleright ensure zero-diagonal

$[G_{obs}]_{ij} \leftarrow 0, \text{ when } [G_{obs}]_{ij} < Q_\alpha(G_{obs})$

Compute eigendecomposition U, Λ of G_{obs}

$\lambda^+ \leftarrow \max(\lambda^{\max}, 0)$

$\lambda^- \leftarrow \min(\lambda^{\min}, 0)$

$c_s^+ \leftarrow \frac{1-\beta}{\beta} \lambda^+$

$c_s^- \leftarrow \frac{1+\beta}{\beta} \lambda^-$

$c_s \leftarrow \max(c_s^+, -c_s^-)$

$\hat{\Lambda} \leftarrow \Lambda (c_s^{-1} I + \Lambda)^{-1}$

return $U \hat{\Lambda} U^T$

where $Q_\alpha(G_{obs})$ 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 1.3.3.

From the definition of $Q_\alpha(G_{obs})$ 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 Section 1.3. 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 [?], 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.

1.3.2 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 [?] as there is a discrepancy between their code¹ 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 of G_{obs} , U , Λ as

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

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 [?] (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 $\rho(\tilde{G}_{dir}) \leq \beta$ is by $c_s = \min\left(\frac{1-\beta}{\beta}\lambda^+, -\frac{1+\beta}{\beta}\lambda^-\right)$ which also implies that $\rho(\tilde{G}_{dir}) = \beta$ 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

¹<https://compbio.mit.edu/nd/>

made in the original discussion of the parameter β in [?]. Furthermore, we note that if we just want the algorithm to converge, as we discussed before, this is equivalent to $\sigma(\tilde{G}_{obs}) \subseteq (-1/2, \infty)$, 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 1.4. 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.

1.3.3 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 quantify the error in G_{obs} , we can from the following analysis quantify the resulting error in G_{dir} . We shall first discuss the original result from [?], 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 $\rho(G_{obs}) < 1$ and $\rho(G_{obs} + N) < 1$ where N is some noise e.g. from computation or assumptions that does not completely hold.

The original result states that $\|G_{dir} - \tilde{G}_{dir}\|_2 \leq \gamma + \mathcal{O}(\delta^2 + \gamma^2 + \delta\gamma)$ 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

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

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 $\rho(M) \leq \|M\|_2$. 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 below ([?]), and that we shall assume that $\rho(G_{obs}), \rho(\tilde{G}_{obs}) < 1$.

Definition 1.9 (Sub-multiplicative Matrix norm). *A matrix norm $\|\cdot\|$ is said to be sub-multiplicative, if for every $A, B \in \mathbb{F}^{n \times n}$ where \mathbb{F} is either the real or complex field:*

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

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 $G_{dir} - \tilde{G}_{dir}$:

$$\begin{aligned} \|G_{dir} - \hat{G}_{dir}\| &= \left\| G_{obs} (I + G_{obs})^{-1} - \hat{G}_{obs} (I + \hat{G}_{obs})^{-1} \right\| \\ &= \left\| -\sum_{k \geq 1} (-G_{obs})^k + \sum_{k \geq 1} (-\hat{G}_{obs})^k \right\| \\ &\leq \sum_{k \geq 1} \left\| G_{obs}^k - (\hat{G}_{obs})^k \right\| \\ &\leq \sum_{k \geq 1} \sum_{i=1}^k \binom{k}{i} \|N\|^i \|G_{obs}\|^{k-i} \\ &= \sum_{k \geq 1} \sum_{i=1}^k \binom{k}{i} \gamma^i \delta^{k-i} \\ &= \sum_{k \geq 1} \left((\gamma + \delta)^k - \delta^k \right) \\ &= \frac{\gamma + \delta}{1 - \gamma - \delta} - \frac{\delta}{1 - \delta} \\ &= \frac{\gamma}{(1 - \gamma - \delta)(1 - \delta)} \end{aligned} \tag{1.9}$$

where in the second to last inequality, we assume that $\gamma + \delta < 1$ as then both $\sum (\gamma + \delta)^k$ 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 $\delta \rightarrow 1$. 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:

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

assuming that $\gamma, \delta > 0$ corresponding to neither N nor G_{obs} is the zero matrix in which case the above analysis is 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 [?], which is very useful when doing matrix perturbation analysis.

Theorem 1.10 (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*

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

The proof of the above is simple through direct computation. Hence, we continue

to once again consider the difference $G_{dir} - \tilde{G}_{dir}$

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

where the third equality follows from Theorem 1.10. 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.

$$\|G_{dir} - \tilde{G}_{dir}\| \leq \|N\| \left\| (I + G_{obs})^{-1} \right\| \left\| (I + G_{obs} + N)^{-1} \right\| \quad (1.10)$$

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 1.9. Equation 1.10 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 Section 1.3 we discuss the error from using mutual information in the case of a multi-variate Gaussian.

From Equation 1.10, by another application of Theorem 1.10, we find the relative error in general to be bounded as follows

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

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 [?], 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 $\sigma(I + G_{obs}), \sigma(I + \tilde{G}_{obs}) \subseteq (1/2, \infty)$ implies $\sigma\left((I + G_{obs})^{-1}\right), \sigma\left((I + \tilde{G}_{obs})^{-1}\right) \subseteq (0, 2)$ we infer that

$\left\| (I + G_{obs})^{-1} \right\|_2, \left\| (I + \tilde{G}_{obs})^{-1} \right\|_2 \leq 2$. Using this with the above equivalence on the Euclidean norm with Equation 1.10, we conclude that

$$\begin{aligned} \left\| G_{dir} - \tilde{G}_{dir} \right\|_F &\leq 4\sqrt{d} \|N\|_F \\ \left\| G_{dir} - \tilde{G}_{dir} \right\|_{\max} &\leq 4d \|N\|_{\max} \end{aligned} \quad (1.11)$$

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.

1.3.4 KDE methods

Det her mangler lige

1.3.5 B-splines

B-spline - non-uniform marginals

[?] Cox-de Boor recursion formula

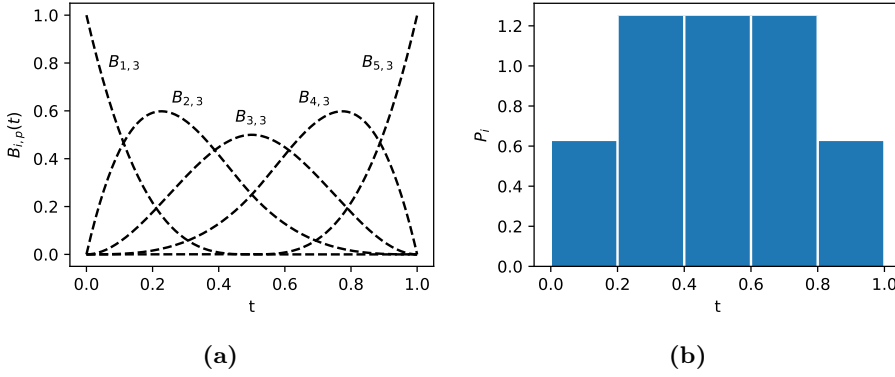
$$B_{i,0}(t) = 1, \quad t \in [t_i, T_{i+1}) \quad (1.12)$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t) \quad (1.13)$$

Have the useful property that $\sum_i B_{i,k}(t) = 1$ for all k and t . However, as we will be using this when trying to estimate mutual information from a copula, we have a very specific problem, namely, the marginals are not discrete uniform. $i \in \{1, \dots, n\}$ basis functions

Let \mathbf{X} be a random vector of d , i.i.d uniformly distributed random variables X_j . Then the random variable corresponding to the probability mass assigned to bin i , P_i , is given by

$$P_i = \frac{1}{d} \sum_{j=1}^d B_{i,p}(X_j)$$



thus, the expected value of P_i is (independently of d)

$$\mathbb{E}[P_i] = \mathbb{E}[B_{i,p}(X_j)] = \int_0^1 B_{i,p}(x) dx$$

Ideally, this should be equal for all i as we want the marginal to be uniform as per definition of a Copula density. However, this is not the case. In fact,

In particular, we can calculate the expected probability mass for each bin i as

1.3.6 M-splines

M-spline - need to normalize distributions

[?] a set of n basis functions indexed by i . Letting the order of the basis function equal $p + 1$ such that the basis functions are polynomials of degree p . Define knots to be $t_1 = \dots = t_k = 0$ and $t_{n+1} = \dots = t_{n+k} = 1$. furthermore, we choose the remaining $n - k$ (interior) knots to be equally distributed on $[0, 1]$ i.e. if 3 interior points, they are $1/4, 1/2$ and $3/4$ respectively. (slightly reformulated to agree with the notation for B splines)

$$M_{i,0}(t) = \frac{1}{t_{i+1} - t_i}, \quad t \in [t_i, t_{i+1}) \quad (1.14)$$

$$M_{i,k}(t) = \frac{k((t - t_i) M_{i,k-1}(t) + (t_{i+k} - t) M_{i+1,k-1}(t))}{(k - 1)(t_{i+k} - t_i)} \quad (1.15)$$

such that $M_{i,p}$ is a degree p polynomial on $[t_i, t_{i+p+1})$

it follows that each of these $M_{i,p+1}$ basis splines integrate to 1 which, as we saw before is important for uniform marginals. In particular, if we instead let

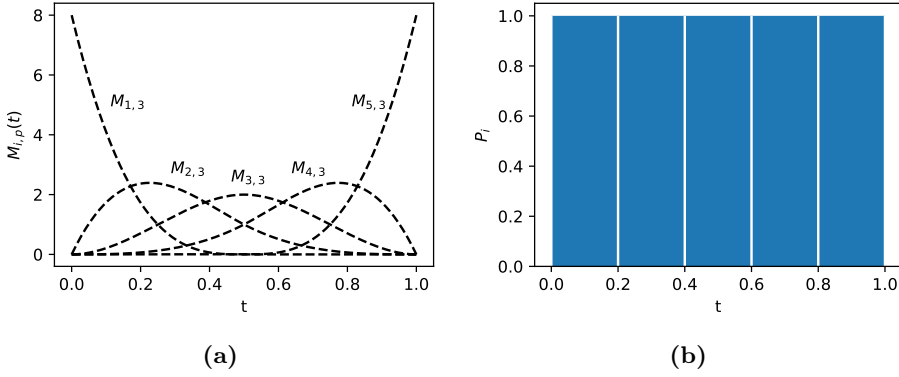


Figure 1.3: P_i is area of each rectangle i.e. 0.2.

the basis functions be $\tilde{M}_{i,p} = \frac{1}{n} M_{i,p}$, The marginal is uniform. However now there is the downside that basis functions do not sum to 1 as with the B-spline. Namely, points near the boundary are weighted more than points on the inside as is illustrated below

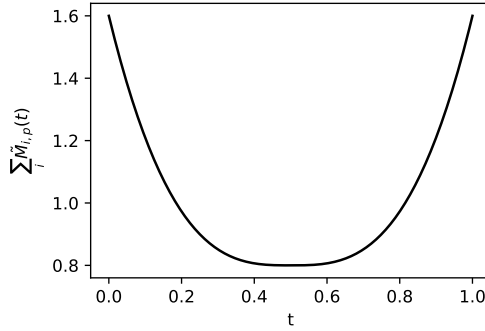


Figure 1.4

1.3.7 A better spline

In theory, we could make a set of splines ourselves, that has both of the desired properties of B-splines and M-splines. I.e. a set of piecewise polynomials, which we shall denote $Q_{i,p}$, that has both the property that

$$\sum_{i=1}^n Q_{i,p}(t) = 1$$

and

$$\int_0^1 Q_{i,p}(t) dt = \frac{1}{n}$$

Actually, both M- and B-splines satisfies this for $p = 0$ as they are then both piecewise constant and non-zero on disjoint intervals. From this, one can the construct such sets of $Q_{i,p}$ in the following way. Namely, start with the piecewise constant functions, $Q_{i,0}$ depicted in Figure 1.5 as boxes in the case $n = 5$.

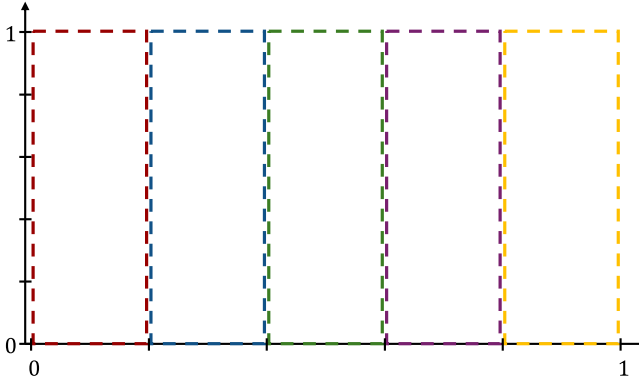


Figure 1.5

A better spline

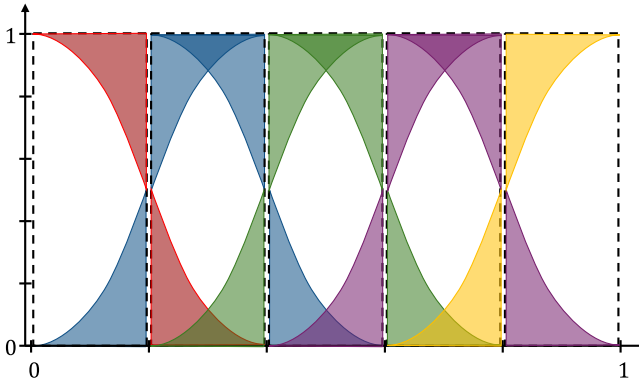


Figure 1.6

Cut off function has derivative 0 at top and is halfway down at the sides

This example does however not work well for many bins, in particular, due to the splines not overlapping that much i.e. a poor regularization on the smoothness.

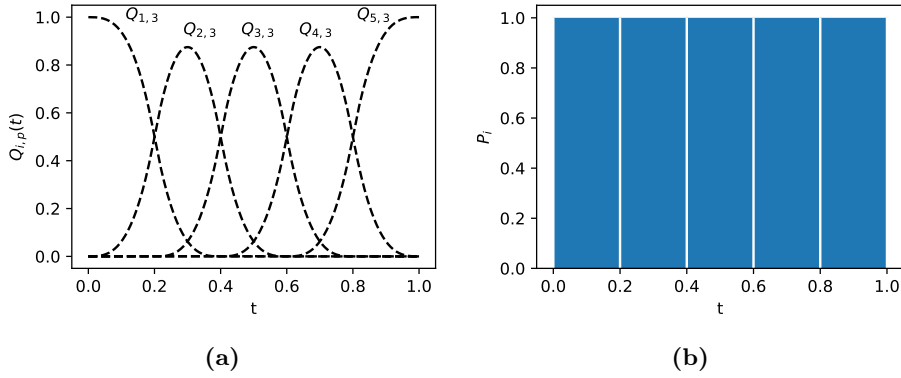


Figure 1.7: P_i is area of each rectangle i.e. 0.2.

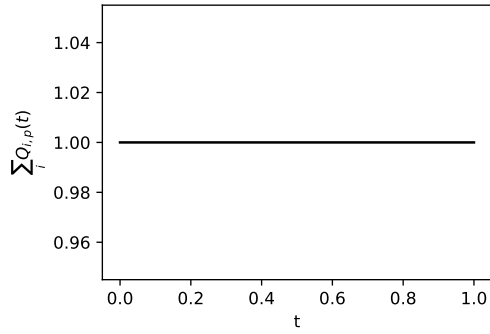


Figure 1.8

However, the method used for constructing this set of splines can be extended to bleed into multiple neighboring bins just as for B- and M-splines.

1.3.8 Naïve KDE

Reflection (ikke så god)

1.3.9 Boundary corrected KDE

Den fra artikel i kina

We shall also try the method briefly presented in [?], which is based on [?]. It is a first order kernel but only specified in one dimension and initially only for a lower bound. The

Furthermore, in [?], Jones and Foster propose an extension to handle the possible negativity of the estimate density.

Also, in R we see a copula method which is a $[0, 1]$ domain method using gaussian copula as kernel such that it is reflexive in x and kernel point X_i . Could also be used and we have implemented this as the above using the products of kernel. [?]

using the one from china, we get that for large correlation, the algorithm on 500×500 does not do well (refer to image on convergence when accurate PDF -> improved scheme by adaptively increasing the mesh size if one gets a large MI)

<https://search.r-project.org/CRAN/refmans/evmix/html/bckden.html>

Hvad med denne: [?]

1.3.10 Diffusion based KDE