Dynamic Bayesian Networks for Business Data.

Student: Cornelis S.L. Rook,

Internal UoG assessor: M.A. Grzegorczyk,

Internal second UoG assessor: W.P. Krijnen

Faculty of Science and Engineering

August 2025

rijksuniversiteit
groningen

Contents

1 Introduction

Bayesian networks are used in a variety of fields. They find many applications in medical diagnosis and healthcare processes, as well as in genetics, environmental sciences, engineering and education (Russell & Norvig, 2022). However, although more commercial fields, such as transportation, logistics and marketing, have also seen applications of Bayesian networks, the field of sales and e-commerce has seen far less exploration and use of this increasingly popular tool. With the rise of the Internet of Things (IoT) (Hassan, Khan, & Madani, 2018), more data-driven approaches for businesses have become increasingly interesting. To this end, this thesis aims to deliver a detailed approach that suits the conditions of real-life business operation data. The goal of the resulting network is to provide insight and prediction methods that guide the business on how to manage operations within the business in a more effective way. There are often many variables that exist within a business, and some of these represent or compose key points of interest that need to be improved. However, it is very difficult to manually qualify, quantify or assess which business variables relate to other business variables. A Bayesian network can therefore be a very interesting tool for businesses to make informed and more efficient decisions. This places the Bayesian network in the role of an expert system that can provide the business with invaluable insights on key performance indicators (KPIs) and their relationships to other variables.

More specifically, the reason a Bayesian network approach as an advisory expert system could prove to be valuable in a business domain is because there are many different key performance indicators that the business might aim to improve on. Moreover, such a network can give a clear and intuitive overview of which empirical relationships exist between these KPIs and any of the other business variables among a potentially huge collection of variables. When there might be many variables in play for a KPI, the Bayesian network can greatly help to inform the business on which variables are the most relevant to focus on with regard to improving their KPIs. In this way, the Bayesian network can be used to enhance decision-making for the business of interest of this paper.

Namely, the data for this research are supplied by a Dutch business called Custom Decks, which primarily focuses on the sale of skateboard decks and parts. The business collects data from various sources in an endeavor to monitor its performance on KPIs such as turnover, web shop visitors and social media traction. The data are programmatically collected on a daily basis via the Google, Instagram APIs, and the web shop's back-end API. In addition, data are collected from KNMI (the Royal Dutch Meteorological Institute) in order to investigate the relationship between weather conditions and skateboard sales, as skating is mostly an outdoor sport. Although it is likely that many relations exist between variables that are collected from the same source, the existence of relationships between variables from different sources is particularly interesting. These relationships highlight the connections between different facets of the business and can provide valuable insights into optimizing KPI performance. With the Internet of Things and the growing body of data that comes with it, Bayesian networks could play an increasingly relevant role in improving strategies for businesses.

In contrast to typical static Bayesian networks, a dynamic approach can be used that leverages the temporal effects of time-series data like the data used in this study. In other words, previous observations are used to inform the likelihood of or even predict contemporary observations. A Bayesian network can capture dynamic relationships by adding columns to the data that contain lagged values of the variable that is to be dynamically modeled. This makes introducing the temporal element of a dynamic model rather straightforward. However, in the context of the application at hand (in which a row of data represents a single day generally speaking), a lag of a single day might be neither sufficient nor accurate to capture meaningful dynamic relationships. An example of this in light of the IoT is social media. A social media post on some platform might go viral and spark a sudden increase in other metrics on the same day or the following day. However, the impact of such an event does not stop after one day. Even more so, the impact of multiple posts or even a single one might accumulate over time within a limited time window before the sum of these effects starts to diminish. However, with a relatively small data set at hand that contains a few hundred rows, adding several lags for most of the variables might result in a very large set of variables where conditional probabilities have to be determined from small or sparse sets of combinations of parent states, which can lead to problems in parameter learning as well as slow down computational speed. Therefore, a preliminary coherence analysis will be performed to deduce which lags contain the most meaningful information with respect to other variables. By using this informed lag approach, important relationships between KPIs and other business variables can be determined by the structure learning algorithm while ensuring a limited number of nodes are used. All in all, this sets up a framework that can be used to ultimately model KPIs with a bounded number of nodes and networks. This will be done by first discretizing the data, after which the networks are learned from the discretized data.

Another aspect in this study is the application of multiple discretization methods, for which the resulting networks are compared as well. Discretization was deemed necessary because the data appeared to not adhere to a Gaussian distribution, as required for continuous Bayesian networks (Scutari & Denis, 2021). With regard to using a mixed or strictly discrete Bayesian network, there are several considerations that need to be made. With the IoT, a virtually arbitrary number of variables of various natures might be introduced into the network. Moreover, not every variable is continuous, as some are simple, discrete labels. This also adds complexity to the network, as nodes representing a continuous variable cannot point towards discrete variables (Scutari & Denis, 2021). To address this problem, all variables will be subjected to discretization. This eliminates the requirement of normality of distributions of continuous nodes entirely and improves performance at the cost of some amount of information (Grzegorczyk & Husmeier, n.d.). Some common practices for discretizing data include splitting the data based on intervals, quantiles, or pairwise mutual information (Hartemink, 2001). While the advantage of interval or quantile methods is that the resulting discretized data has some regular properties such as equally sized intervals or equally sized batches of data for each value, their general drawback is that the boundary values of these discretizations are arbitrary and do not take into account the spread of the data. Although the Hartemink discretization typically performs better than interval or quantile discretization, an additional discretization approach was employed in this study, in order to better reflect the distribution of individual variables. Namely, the data were discretized using a medoid approach according to the partitioning around medoids algorithm (PAM) (Schubert & Rousseeuw, 2021). The main advantage of this approach is that patterns specific to the spread of a single variable are more faithfully reflected in the resulting discretization. All in all, comparing the quality of networks learned from different discretizations of the data will be one of the core components of this study.

After the data are discretized, the structure of the Bayesian network needs to be learned. There are several types of algorithms that can do this. The most notable types are score-based algorithms and constraint-based algorithms, which provide fundamentally different approaches to structure learning. However, other approaches based on local discovery of the undirected network, mutual information or hybrids are also applied, often depending on the data (Scutari & Denis, 2021). Given the new application in this paper, each of these algorithm types will be explored in order to assess their suitability within the scope at hand. This is particularly interesting given the temporal component of the data as well as the dynamic approach. Therefore, the Hill-Climbing (Russell & Norvig, 2022), pc.stable (Colombo & Maathuis, 2015), Max-Min Hill-Climbing (Tsamardinos, Brown, & Aliferis, 2006) and ARACNe algorithm (Margolin et al., 2006) will be applied and scored using various metrics.

For this paper, the aim is to propose a comprehensive approach for discovering dynamic Bayesian networks when using time-series business data that revolve around KPIs. Firstly, the theory on Bayesian networks is discussed. In particular, several approaches to structure learning will be discussed, including score-based and constraint-based algorithms, among others. The theory section will be concluded by presenting the evaluation metrics used for the Bayesian networks. Secondly, the preliminary analysis and setup are discussed. This will consist of an exploratory application of the structure learning process used in this study on a benchmark data set for which the true network is known. This provides insights into the effectiveness of the processes used. In addition, a cross-correlation analysis of the variables is performed to deduce appropriate time aggregates of the data. This was complemented by different types of edge blacklists so that dynamic networks could be constructed in one of two ways. Given that dynamic Bayesian networks use time-lagged instances of variables to represent prior observations as nodes, one type of blacklist allows edges from lagged variables to contemporary ones alongside edges between contemporary variables, whereas the other does not allow edges between contemporary nodes. These blacklists come with different qualities and properties for the networks. Hence, the resulting networks can be interpreted in separate ways and have different merits. With regard to discretization, the PAM algorithm will also be presented and discussed in light of more conventional discretization methods.

Lastly, this paper aims to answer the following research questions.

- Which discretization is best suited for our sales application?
- Which structure learning algorithm is best suited to learn networks on time-series business data?
- What is the impact of the different blacklists on the effectiveness of the networks?

1.1 Related work

Currently, most applications of Bayesian networks lie within medical applications, research, and risk assessment. Only a handful of papers have been presented in recent years that cover an application of Bayesian networks within the scope of sales or business yields from a broad perspective. Some examples are Verbraken, Verbeke, and Baesens (n.d.) and Horita and Yamashita (2019). These articles display applications of Bayesian networks that optimize specific aspects of a business, such as customer analysis and churn, rather than optimizing a collection of KPIs. Other applications in this field include ones that do not employ Bayesian networks directly as a type of expert system, but rather use Bayesian inference methods when training neural networks (Wang, Wang, & Wang, 2009), (Heskes, Spanjers, & Bakker, n.d.), (Ragg, Menzel, Baum, & Wigbers, 2002)

Moreover, very few articles present applications of dynamic Bayesian networks in the domain of sales and commerce. An example might be Zhao, Xie, and West (2016), where a dynamic Bayesian network is used to produce portfolio analyses, but this application study limits dynamic modeling to an autoregressive approach. To this end, the present study probably introduces a novel application of dynamic Bayesian network modeling.

Substantial efforts have been made in the development and analysis of various dynamic Bayesian network models. The arguably seminal works of Marco Scutari have been used extensively to produce dynamic Bayesian networks. In particular, Scutari and Denis (2021) and Scutari (n.d.) provide an extensive background and analysis on Bayesian networks. These works elaborately discuss real-world applications and limitations of Bayesian networks, as well as extensive analysis on various structure learning algorithms, among many other technical details. This is corroborated by the detailed theoretical exercises displayed in Grzegorczyk and Husmeier (n.d.) and Kamalabad and Grzegorczyk (n.d.), which predominantly contribute to the literature on dynamic Bayesian networks in particular. Naturally, this literature on dynamic Bayesian networks served as a major inspiration for the research and provided this research with several theoretical insights.

The use of medoid clustering is frequent in the domain of deep learning and neural networks (Panda, Mousa, & Hassanien, 2021), (Salama & Freitas, 2014). However, the application of this clustering algorithm in the context of Bayesian networks appeared to be very limited. Some instances that used this type of clustering used it for different purposes compared to its purpose in this study. In particular, the medoid clustering method was used for feature engineering in Panda et al. (2021) and used inside structure learning processes (rather than preprocessing) for Bayesian network classifiers (Salama & Freitas, 2014).

All in all, our study distinguishes itself from previous work on Bayesian networks by presenting a dynamic approach in combination with a novel application setting. An extensive preliminary analysis on the data, as well as on some initial networks, elicits interesting differences between different approaches to modeling (dynamic) Bayesian networks, as well as some key similarities. Subsequently, the results of the models are compared across algorithms with a variety of metrics such as the BIC and Brier scores. In addition, the results are also cross-validated and bagged to investigate the reliability and stability of the models more thoroughly.

2 Method

2.1 Bayesian networks

Bayesian networks (BNs) are graphical models that can be used to represent a set of n random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. Each of these random variables represents some column from the measured data and hence represents some quantity of interest. In a BN, nodes are used to represent these random variables and (directed) edges between these nodes are used to denote a dependency structure between the respective nodes. However, one strong condition for creating valid BNs is that they have to be acyclic, meaning that loops where the directions within a sequence of edges lead to a node already present in the sequence are prohibited. We will denote such a directed acyclic graph (DAG) as symbol G. Furthermore, the set of edges will be denoted by $\mathbf{I}$, so that in combination with the set of random variables $\mathbf{X}$, we can denote a particular representation of the dependencies within the data as $G(\mathbf{X}, \mathbf{I})$. Since edges imply conditional dependence between nodes, this dependence can be expressed by a Bayesian probability in the case of a single edge between two nodes. To properly define these dependencies, the definitions of nodes and edges will be extended and stated more formally:

Definition 2.1. *For a graph G, with nodes $X_1, ..., X_n \in \mathbf{X}$, an edge from the set $\mathbf{I}$ connecting nodes X_i and X_j with direction $X_i \to X_j$ is denoted by I_{ij}, such that $I_{ij} \in \mathbf{I}$.*

Note that not every pair of nodes is required to have an edge, since the interpretation of the presence of an edge means that there is a dependency between the two nodes. Thus, only variables that do have an association will be connected via an edge, and the absence of an edge implies conditional independence. Moreover, the existence and direction of this edge indicates and dictates the direction of the conditional dependency. If directed edge I_{ij} connects nodes X_i and X_j, it means that X_j is conditionally dependent on its so-called parent X_i. To make this more general, we introduce the following definition.

Definition 2.2. *A node X_i is said to be in an ancestor set $An(X_j)$ of X_j if the graph holds at least one sequence of one or more edges starting from X_i such that this sequence of edges starts with I_{ik} and ends with I_{lj} for some distinct $i, j, k, l \in \{1, 2, ..., n\}$.*

As can be seen from this definition, multiple nodes can belong to the ancestor set of a node. These nodes are not required to belong to the exact same sequence. There can be different nodes X_i and X_{i+1} that do not have an overlapping sequence. This means that a node can have ancestors that do not share ancestors. In particular, we will consider a special kind of ancestor. Namely, a direct ancestor of a node.

Definition 2.3. *A node X_i is said to be in the parent set $Pa(X_j)$ of X_j if the node is attached to an edge I_{ij} that directly points into X_j. This is denoted as $X_i \to X_j$.*

To give an example, the conditional dependence implied by an edge can be explained intuitively by considering a small graph such as $X \to Z \leftarrow Y$, where X, Y and Z represent nodes. As we can see, there is no edge connecting X and Y directly, which means that $P(X|Y) = P(X)$ and $P(Y|X) = P(Y)$ for any realizations of X and Y, implying independence. However, if we consider a realization of Z, things get more complicated. Imagine that both X, Y and Z can have two different realizations, 0 and 1 and that the probability of 1 occurring in Y increases the likelihood of 1 occurring in Z as well. Then, $P(Y = 1|Z = 1, X = 1) < P(Y = 1|Z = 1)$, which is trivially true because the left-hand side is conditioned on a subset of the realizations on which the right-hand side is conditioned. Now we see that there is some interaction between variables X and Y given a realization in their shared child node. Naturally, the probability of $P(Z = z|Y = y)$ or $P(Z = z|X = x)$ is empirically determined because the dependency structure is unknown beforehand.

Continuing with our mathematical definitions, we can represent the probability that some state x_{j_k} of some random variable X_j occurs, where $x_{j_k} \in \{x_{j_1}, ..., x_{j_{r_j}}\}$ and r_j represents the number of possible states of variable X_j. Namely, within graph G, where an edge originates from X_i and directs to X_j, a realization of the random variable X_j depends on the realization of X_i:

$$P(X_j = x_{j_k}|G) = P(X_j = x_{j_k}|X_i = x_{i_l}) \tag{1}$$

Now, considering all r_j possible states of the child node X_j, this constitutes a probability density function as shown below in equation (2). That is, given a parent state l, the states of the child node collapse into a probability distribution that is conditioned on this said parent state.

$$\sum_{k=1}^{r_j} P(X_j = x_{j_k}|X_i = x_{i_l}) = 1 \tag{2}$$

That is, the probabilities that these r_j states occur in X_j when considering some arbitrary realization l of the parent node are stochastic and represent a probability distribution. Note that this example only considers the local distribution for the case where a node has one parent. We can generalize this example to specify the global distribution. This is done by producing a factorization of local distributions for all nodes given their parent set and corresponding distribution parameters for the child node. All in all, this leads the global distribution to be specified by:

$$P(\mathbf{X} \mid G, \Theta) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i), \theta_{X_i}), \tag{3}$$

where θ_{X_i} specifies the parameters of the probability distribution of X_i, which is a conditional probability distribution when the node has a non-empty parent set. Otherwise, it is defined simply by its prior distribution. Gaussian distributions are assumed for continuous variables in this case (Grzegorczyk & Husmeier, n.d.). In general, alternative distributions can be specified for a node that represents a continuous variable, but only if the structure of the graph is known already (Scutari & Denis, 2021). However, the application at hand will deal with discretized data, in which case a multinomial distribution, often also called the categorical distribution, is assumed and models the probabilities of states simply as their respective relative frequencies, as shown in equation (4). This choice was motivated by the fact that it does not require prior knowledge of the distributions, which makes it more suitable for a novel application. Furthermore, measuring the normality of the variables using a Shapiro-Wilk test (Shapiro & Wilk, 1965) resulted in no variable being assessed as approximately normally distributed (Rook, 2025). Hence, using a continuous BN would not have been a suitable approach.

Defining the multinomial distribution for a discrete random variable, we again denote a set of unique realizations of X_i as $x_{i_k} \in \{x_{i_1}, ..., x_{i_{r_i}}\}$ with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{r_i})$, where $\sum_{k=1}^{r_i} \theta_k = 1$, so that

$$P(x_{i_1} = N_{i_1}, \ldots, x_{i_{r_i}} = N_{i_{r_i}} \mid \boldsymbol{\theta}) = \frac{N_i!}{N_{i_1}! \cdots N_{i_{r_i}}!} \prod_{k=1}^{r_i} \theta_k^{N_{i_k}}, \tag{4}$$

where $N_i = \sum_{k=1}^{r_i} N_{i_k}$ denotes the total number of observations N for variable X_i and N_{i_k} denotes the number of observations with value $k \in \{1, \ldots, r_i\}$ for said variable.

Typically, the graphs are constructed from scratch, meaning no initial edges, often using forward selection algorithms, or algorithms considering local or global neighborhoods for edge mutation. At every iteration, a mutation (addition, removal or reversal of an edge) is considered and accepted or rejected. Often applied methods are greedy search algorithms, or Markov-like processes, such as the Metropolis-Hastings algorithm, which incorporate randomness such that the algorithm is less likely to get stuck on local optima, but there exist other methods as well. Among algorithms, the probability of accepting a mutation is defined by some (probabilistic) metric. Most algorithms use metrics such as posterior log-likelihoods, information scores (AIC/BIC/BGe), or entropy measures to quantify how likely the graph is to properly represent the data, given the set of edges and directions at the current iteration. This quantifies whether the mutation improved the graph. After the algorithm is terminated by some stopping criteria, the final graph is selected as the proposed network.

Fundamentally, structure learning algorithms aim to find the graph G that is most likely to explain the data. This is expressed as $P(D|G)$. Using the multinomial distribution to model conditional distributions in a discrete Bayesian network, we can derive a meaningful prior to model likelihood $P(D|G)$. In particular, the Dirichlet distribution is the conjugate prior of the multinomial distribution. Therefore, the posterior is also Dirichlet, but with updated parameters. It is because of the conjugacy that the integral

$$P(D|G) = \int P(D|\theta, G)p(\theta|G)d\theta \tag{5}$$

has a closed-form solution. Let us also define the Dirichlet prior. Let X_i be a discrete random variable with r_i possible values, $Pa(X_i)$ be the parent nodes of X_i with q_i total configurations. Then, for each parent configuration $j \in \{1, \ldots, q_i\}$ (note that j now indexes a parent configuration rather than a random variable), the conditional probabilities can be modeled as

$$\theta_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$$

where

$$\theta_{ijk} = P(X_i = k|Pa(X_i = j), \quad \sum_{k=1}^{r_i} \theta_{ijk} = 1$$

$$p(\theta_{ij}) \;=\; \frac{1}{B(\alpha_{ij1}, \ldots, \alpha_{ijr_i})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1}, \tag{6}$$

where B is the multivariate Beta function,

$$B(\alpha_{ij1}, \ldots, \alpha_{ijr_i}) = \frac{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ij})}, \quad \alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}, \tag{7}$$

where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$. If we let $N_{ijk} = \#\{X_i = k, Pa(X_i) = j\}$, the likelihood is multinomial with:

$$P(D|\theta, G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$

Then, by conjugacy, the posterior for θ_{ij} is Dirichlet:

$$p(\theta_{ij} \mid D, G) = \mathrm{Dir}(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}).$$

Returning to the integral in equation (5), one would have per node X_i and parent configuration j

$$\int P(D|G, \theta_{ij})\, p(\theta_{ij}|G)\, d\theta_{ij} = \frac{1}{B(\alpha_{ij1}, \ldots, \alpha_{ijr_i})} \int \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1}\, d\theta_{ij}. \tag{8}$$

Then, note that the integral on the right-hand side is a Dirichlet normalizing constant, and hence can be written as

$$\int \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1} \, d\theta_{ij} = B(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i}).$$

Substituting this into equation (8), results in

$$\int P(D|G, \theta_{ij}) \, p(\theta_{ij}|G) \, d\theta_{ij} = \frac{B(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i})}{B(\alpha_{ij1}, \ldots, \alpha_{ijr_i})}.$$

Finally, we apply equation (7) and factorize the whole to account for all nodes and parent configurations implied by G. This means that the likelihood of the data D, given a graph G, as defined in equation (5), can be expressed in a closed form as

$$P(D \mid G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{9}$$

where $N_{ij} = \sum_k N_{ijk}$. In the limit of $N \to \infty$, this approaches the BIC score. The proof of this convergence will be provided when discussing the BIC score in the structure learning context in more detail.

2.2 Dynamic Bayesian networks

In addition to regular Bayesian networks that represent the dependency structure of a collection of random variables $\mathbf{X}$, a dynamic Bayesian network (DBN) can be used to reflect temporal dependencies between variables. The setup of a DBN consists of applying a delay of a desired interval size t to the data frame that represents $\mathbf{X}$. This implicitly assumes that the values at some row in the data $\mathcal{D}$ represent averages of the values attained in this interval. Otherwise, it would imply that the variables can impact each other without any time passing (Scutari & Denis, 2021). Therefore, the data should either already represent aggregated values or be aggregated otherwise. For time interval t_i, let us denote the resulting set of random variables by $\mathbf{X}_{T_s}$, where $T_s \subset \{t_1, t_2, ..., t_d\}$ represents a set of delays. Note that $\mathbf{X} \subset \mathbf{X_t}$, assuming $t > 0$, where t represents the shift in terms of number of rows in $\mathcal{D}$ to create the delayed variables. With the additional variables, some restrictions on the dependencies are typically imposed by means of a blacklist. Typically, the restrictions are some combination of the following:

1 No edges between nodes that both do not represent the current or non-delayed instance of a random variable.

2 Edges can only follow the direction of time.

Other than more specific restrictions related to the context of the application, the restrictions above restrict the edge space in structure learning to one that can only provide sensible and usable edges. For instance, omitting restriction **1** will allow the structure learning algorithm to set edges between pairs of delayed variables. This is often not desirable, because one is usually interested in the effects on the non-delayed nodes. That is, the perceived influence of a past observation on a current observation is more valuable and generally more insightful than the perceived effects between delayed observations, especially when using the network for prediction. Another reason for this is that such co-occurring effects would likely show up between the nodes representing the current observation as well because these nodes represent the same, albeit shifted, patterns.

The second restriction limits the search space to edges that only follow a proper chronological direction, as it would not make sense in the real world that current observations would influence delayed observations. Therefore, it is important to blacklist these edges that point backward in time, because that means that any association found between two nodes results in a proper direction because of the temporal nature of the data. More specifically, in order to really leverage the temporal component of $\mathbf{X_t}$, it is argued that edges should also be prohibited between all nodes $X \in \mathbf{X}$ (Scutari, n.d.). However, the latter restriction is not always fully employed, because the edges between the nodes of $\mathbf{X}$ represent edges that would otherwise likely have been found in the regular BN. In that case, the edges should follow the direction of time, or live between the original contemporary nodes in $\mathbf{X}$. However, using a blacklist that fully enforces rule **2** forces the edges in the network to only point from past to present. This can be useful when co-occurring edges are not of interest or illogical in practice. With these two types of blacklists, two different types of DBNs can be constructed.

Without loss of generality, the use $\mathbf{X}_{T_s}$ is omitted in proofs and other theory presented in the theory section. Namely, the regular notation for a set of variables $\mathbf{X}$ will be used by default, unless explicitly talking about delay sets and rolling windows for constructing a DBN-suited data set.

2.3 Structure learning

Given the novel application, one particular interest of this research is to find an algorithm that provides not only the most insightful results, but also stable results. The metrics considered for this will be discussed in more detail later. For now, the algorithms will be presented and discussed. In the study at hand, all structure learning was started with empty graphs. These graphs have all the nodes, but no edges, meaning that the algorithm can leverage any of the nodes, but has to determine which edges are to be included or excluded. Starting with the Hill-Climbing algorithm shown in algorithm 1, this score-based algorithm constitutes the simplest algorithm in this paper and is a typical greedy search algorithm.

Algorithm 1 Hill-Climbing Algorithm

```
1: function Hill-Climbing(emptygraph, data) returns a state that is a local maximum
2:     current ← problem.Initial
3: while true do
4:     neighbor ← a highest-valued successor state of current
5:     if Value(neighbor) ≤ Value(current) then
6:         return current
7:     end if
8:     current ← neighbor
9: end while
```

In this algorithm, the input is an empty graph along with the data set from which to learn the structure. In line 4 of the algorithm, all of the possible neighboring graphs are considered. Given some graph G, a neighboring graph G^* is one that is only one operation away, where an operation can be one of edge addition, deletion or reversal. Generally speaking, this means that the search space is of order $O(N^2)$, where N is again the number of random variables. This can be seen from the fact that for one node in an empty graph (which is one out of N) there are $N-1$ other nodes with which it can be connected. However, since an edge cannot be added in the reverse direction when there is an existing edge between the two nodes, the total number of possible edges becomes $\frac{N*(N-1)}{2}$. Similarly, this quadratic order holds for edge reversal and deletion. Therefore, the order of the search step in line 4 is of order $O(N^2)$, making the algorithm less suited for data sets with many variables. Next, after the search is performed, the greedy selection is made by always selecting the operation that yields the best score. The Bayesian Information Criterion (BIC) was used to score the structure at every iteration. Hence, the neighboring graph with the lowest BIC score is selected before going to the next iteration. However, if no such better neighbor can be found, the algorithm terminates and the previous network is returned as the best model. Note that although no parameters are learned yet in this phase of Bayesian network learning, conditional probability tables (CPT) are used to represent the current network parameters. Using these CPTs, the BIC score can be computed to weigh the goodness of fit against the model complexity, as displayed in equation (10) (Wit, van den Heuvel, & Romeijn, 2012). In this equation, $\hat{L}$ represents the maximum likelihood function under the current dependency structure implied by the graph G that is to be scored. The value k represents the number of parameters, which corresponds to the number of entries in the CPT. The number of entries in a CPT is determined by the product $r_i \prod_{X_j \in Pa(X_i)} r_j = r_i q_i$, where r_i, denotes the number of possible values for the variable X_i and q_i denotes the number of parent configurations, as discussed earlier. Lastly, N represents the number of data points. From the equation, it can be seen that a graph that produces a higher likelihood will have a greater score, but that the score is decreased or penalized by the amount of parameters included in the model. Note that this penalty is scaled by the number of observations, which typically is a fixed number, meaning that only $\hat{L}$ and k can vary between different iterations. Hence, this algorithm aims to find a network that provides the best trade-off between improving the log-likelihood and diminishing the penalty obtained from the amount of entries in the CPTs.

$$\text{BIC} = \log \hat{L} - \frac{k}{2} \log(N) \tag{10}$$

In addition to using the BIC score during the HC algorithm, the score is widely used to score networks found by any type of algorithm. This is done because the BIC score approaches the so-called BDe score in the limit of the number of observations in the data. The BDe score can be interpreted as the marginal likelihood of the data, given a network, i.e. $P(D|G)$. Under certain conditions such as a BN being discrete or Gaussian, this quantity can be computed in closed form. However, the BIC value is much simpler in use because it requires no distributional prior and hyperparameters and saves computation time required, which can be useful when methods such as model averaging are used. Because the convergence of the BIC score to the BDe score is central to our argument in thesis, a proof of this convergence will be provided for discrete networks.

Let us again state the meaning of each variable for clarity. Let node X_i with r_i possible values and q_i parent configurations and N_{ijk} be the number of times $X_i = k$ while $Pa(X_i)$ has configuration j. Let the parameters $\theta_{ijk} = P(X_i = k|Pa(X_i) = j)$ have a Dirichlet prior $\text{Dir}(\alpha_{ij1}, \ldots, \alpha_{ijr_i})$, and define $\alpha_{ij} = \sum_k \alpha_{ijk}$ and $N_{ij} = \sum_k N_{ijk}$. The marginal likelihood of the data, i.e. the BDe score, is given by equation (9) and repeated here:

$$BDe(G, D) \; = \; \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \; \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}.$$

If we now take the logarithm, we get

$$\ell_{\text{BDe}}(G) = \sum_{i,j} \Big\{ \log \Gamma(\alpha_{ij}) - \log \Gamma(\alpha_{ij} + N_{ij}) + \sum_{k=1}^{r_i} \big[\log \Gamma(\alpha_{ijk} + N_{ijk}) - \log \Gamma(\alpha_{ijk}) \big] \Big\}.$$

Now, the Stirling approximation of the gamma function can be applied. This approximation holds for large values of x and is given by

$$\log \Gamma(x) = \Big(x - \tfrac{1}{2} \Big) \log x - x + \tfrac{1}{2} \log(2\pi) + O(1). \tag{11}$$

By applying equation (11), to $\ell_{\text{BDe}}(G)$ and putting all parts that only depend on α_{ijk} in a constant term C_{ij}, we get for each summand part the following equation:

$$\begin{aligned} \ell_{ij} = \sum_{k=1}^{r_i} & \Big[(N_{ijk} + \alpha_{ijk} - \tfrac{1}{2}) \log(N_{ijk} + \alpha_{ijk}) - (N_{ijk} + \alpha_{ijk}) \Big] \\ & - \Big[(N_{ij} + \alpha_{ij} - \tfrac{1}{2}) \log(N_{ij} + \alpha_{ij}) - (N_{ij} + \alpha_{ij}) \Big] + C_{ij} + O(1). \end{aligned} \tag{12}$$

Because summing N_{ijk} and α_{ijk} over k results in N_{ij} and α_{ij} respectively, and by splitting equation (12) into parts regarding N or α, we get

$$(\alpha_{ijk} + N_{ijk} - \tfrac{1}{2}) \log(\alpha_{ijk} + N_{ijk}) = N_{ijk} \log(\alpha_{ijk} + N_{ijk}) + (\alpha_{ijk} - \tfrac{1}{2}) \log(\alpha_{ijk} + N_{ijk}),$$

$$(N_{ij} + \alpha_{ij} - \tfrac{1}{2}) \log(N_{ij} + \alpha_{ij}) = N_{ij} \log(\alpha_{ij} + N_{ij}) + (\alpha_{ij} - \tfrac{1}{2}) \log(\alpha_{ij} + N_{ij}).$$

Applying these equivalencies to equation (12), let us split equation (12) into parts that have a factor N outside the logarithm as A_{ij} and those that have a factor of α outside of the logarithm as B_{ij}.

$$A_{ij} = \sum_k N_{ijk} \log(\alpha_{ijk} + N_{ijk}) - N_{ij} \log(\alpha_{ij} + N_{ij}),$$

$$B_{ij} = \sum_k (\alpha_{ijk} - \tfrac{1}{2}) \log(\alpha_{ijk} + N_{ijk}) - (\alpha_{ij} - \tfrac{1}{2}) \log(\alpha_{ij} + N_{ij}).$$

For A_{ij}, with large N_{ij}, N_{ijk}, we can use

$$\log(N_{ijk} + \alpha_{ijk}) = \log N_{ijk} + O(1/N_{ijk}).$$

so that with $\sum_k N_{ijk} = N_{ij}$

$$A_{ij} = \sum_k N_{ijk} \log N_{ijk} - N_{ij} \log N_{ij} + O(1) = \sum_k N_{ijk} \log \frac{N_{ijk}}{N_{ij}} + O(1).$$

Similarly,

$$B_{ij} = \sum_k (\alpha_{ijk} - \tfrac{1}{2}) \log N_{ijk} - (\alpha_{ij} - \tfrac{1}{2}) \log N_{ij} + O(1).$$

Now write $N_{ijk} = N_{ij}\pi_{ijk}$, $\sum_k \pi_{ijk} = 1$, so that $\log N_{ijk} = \log N_{ij} + \log_k \pi_{ijk}$. Then

$$\begin{aligned} B_{ij} &= \sum_k (\alpha_{ijk} - \tfrac{1}{2})(\log N_{ij} + \log \pi_{ijk}) - (\alpha_{ij} - \tfrac{1}{2}) \log N_{ij} + O(1) \\ &= \Big(\alpha_{ij} + \tfrac{r_i}{2} - \alpha_{ij} - \tfrac{1}{2}\Big) \log N_{ij} - \sum_k (\alpha_{ijk} - \tfrac{1}{2}) \log \pi_{ijk} + O(1) \\ &= \tfrac{r_i - 1}{2} \log N_{ij} - \sum_k (\alpha_{ijk} - \tfrac{1}{2}) \log \pi_{ijk} + O(1). \end{aligned}$$

Combining A_{ij} and B_{ij} again and grouping the constants results in

$$\ell_{ij} = \sum_k N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \frac{r_i - 1}{2} \log N_{ij} + \underbrace{\Big(C_{ij}^{(0)} + \tfrac{r_i - 1}{2} \log(2\pi) + \sum_k (\alpha_{ijk} - \tfrac{1}{2}) \log \pi_{ijk}\Big)}_{C_{ij}} + o(1).$$

Note that for discrete BNs the term $\sum_k N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$ equals the log-likelihood over all values k being observed for variable X_i with parent configuration j. Summing over all (i, j):

$$\log P(D \mid G) = \sum_{i,j,k} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \frac{1}{2} \sum_{i,j} (r_i - 1) \log N_{ij} + \sum_{i,j} C_{ij} + o(1).$$

Using the law of large numbers and assuming an i.i.d. distribution of the data, N_{ij} can be expressed in terms of the number of observations N by $N_{ij} = c_{ij}N(1 + o(1))$ with fraction $c_{ij} > 0$ resolving to a separate logarithm constant within the configurations of i and j,

$$\sum_{i,j} (r_i - 1) \log N_{ij} = d \log N + O(1),$$

Recall that for parent configuration j of $Pa(X_i)$, q_i was defined as the number of parent configurations, so that summing over i and j can be simplified as $\sum_{i,j}(r_i - 1) = \sum_i q_i(r_i - 1) = d$, which is the total number of free parameters. Thus, the log of the BDe score in the limit of number of observations N is approximated by

$$BIC(G, D) = \log L(\hat{\theta} \mid G) - \tfrac{d}{2} \log N + O(1) \to \log P(D|G) = BDe(G, D),$$

where $\hat{\theta}$ is the empirical estimate of θ.

This concludes the proof of the BIC score. It should be noted that the BIC score is essentially a frequentist approach to estimating the marginal likelihood. In contrast, a fully Bayesian approach would involve computing the marginal likelihood directly by means of the BDe method. The proof above shows that the frequentist approach has merit because it converges to the true value of the marginal likelihood.

The next algorithm to be discussed is the constraint-based pc.stable algorithm (Colombo & Maathuis, 2015). To properly understand the algorithm, the following definitions are presented. For clarity of notation, different nodes are temporarily represented by different capital letters rather than different lower case subscript indices.

Definition 2.4 (Skeleton Graph)**.** *The* skeleton graph *of a Bayesian network G is the undirected graph obtained by:*

- *Removing the directionality of all edges in the network.*
- *Retaining only the presence or absence of an edge between nodes, without regard to its direction.*

Formally, given a directed acyclic graph G $(\mathbf{X}, \mathbf{I})$, where $\mathbf{X}$ is the set of nodes and $\mathbf{I}$ is the set of directed edges, the skeleton graph is the undirected graph $(\mathbf{X}, \mathbf{I}_s)$, where:

$$\mathbf{I}_s = \{\{X, Y\} \mid (X \to Y) \in \mathbf{I} \textit{ or } (Y \to X) \in \mathbf{I}\},$$

where $X, Y \in \mathbf{X}$

Definition 2.5 (Adjacency)**.** *A node X is said to be adjacent to node Y if there exists some edge in the set of edges I connecting the two nodes regardless of direction. The adjacency set of a node X in a graph G is denoted as $adj(G, X)$.*

Definition 2.6 (Complete graph)**.** *A graph $G(\mathbf{X}, \mathbf{I})$ is said to be complete if all nodes in $\mathbf{X}$ are adjacent to each other.*

Definition 2.7 (Unshielded Triple)**.** *An* unshielded triple *in a Bayesian network (or the skeleton of a graph) is a set of three nodes (X, Y, Z) such that:*

- *There exist edges between the pair X and Y and the pair Y and Z, regardless of*
- *There is **no edge** directly connecting X and Z, i.e., $(X, Z) \notin \mathbf{I}$*

Definition 2.8 (V-Structure)**.** *A* v-structure *is a specific configuration of an unshielded triple in a directed Bayesian network. It is a triple (X, Y, Z) such that:*

- *$X \to Y$*
- *$Z \to Y$*
- *There is **no edge** between X and Z, i.e., $(X, Z) \notin \mathbf{I}$*

The node Y is referred to as the collider *because the incoming edges collide at Y.*

Definition 2.9 (D-Separation)**.** *Two nodes X and Y in a Bayesian network are* d-separated *by a set of nodes $\mathbf{S} \subseteq \mathbf{X} \setminus \{X, Y\}$ if all paths between X and Y are* blocked *by* $\mathbf{S}$.

A path is considered blocked *by $\mathbf{S}$ if at least one of the following conditions holds:*

- *There exists a* chain $A \rightarrow B \rightarrow C$ *or* $A \leftarrow B \leftarrow C$, *where* $B \in \mathbf{S}$.
- *There exists a* fork $A \leftarrow B \rightarrow C$, *where* $B \in \mathbf{S}$.
- *There exists a* collider $A \rightarrow B \leftarrow C$, *where $B \notin \mathbf{S}$ and no descendant of B is in $\mathbf{S}$.*

If X and Y are d-separated by $\mathbf{S}$, they are conditionally independent given $\mathbf{S}$, denoted as:

$$X \perp Y \mid \mathbf{S}.$$

Definition 2.10 (Separation Set)**.** *The* separation set *(or* sepset*) for two nodes X and Y in a Bayesian network (or its skeleton graph) is a set of nodes $\mathbf{S} \subseteq \mathbf{X} \setminus \{X, Y\}$ such that:*

- $\mathbf{S}$ *d-separates X and Y, meaning X and Y are conditionally independent given $\mathbf{S}$. Namely, $X \perp Y \mid \mathbf{S}$.*
- *The set $\mathbf{S}$ is minimal, meaning no proper subset of $\mathbf{S}$ satisfies this condition.*

Hence, the separation set is a proper subset $\mathbf{S}$, where all nodes in $\mathbf{S}$ satisfy the d-separation property.

Note that in the definition of a complete graph, for some graph G with N nodes, it implies that all nodes in $\mathbf{X}$ need to have $N - 1$ edges that are directed to or from them to be adjacent to all other nodes. This is a very dense graph and this will be the starting point for the pc.stable algorithm.

Now, the pc.stable algorithm consists of a total of three overarching steps that can be broken down into more detailed steps. In algorithm 2, these overarching steps are used to give a general overview of the algorithm (Colombo & Maathuis, 2015).

Algorithm 2 The PC-algorithm

```
Require: Conditional independence information among all variables in $\mathbf{X}$, and an ordering $\mathbf{V}$ on the variables
1: Adjacency search: Find the skeleton $\mathcal{C}$ and separation sets using Algorithm 3;
2: Orient unshielded triples in the skeleton $\mathcal{C}$ based on the separation sets;
3: In $\mathcal{C}$, orient as many of the remaining undirected edges as possible by repeated application of rules R1-R3 (see text);
4: return Output graph $\mathcal{C}$ and separation sets (sepset).
```

In order to define conditional independence between nodes, some metric needs to be employed that quantifies the amount of dependency between them. Because the current application considers only discrete variables, the log-likelihood ratio can be used to quantify the dependency (Scutari & Denis, 2021). The test statistic assumes the form of

$$G^2(X, Y \mid \mathbf{Q}) = \sum_{x \in X} \sum_{y \in Y} \sum_{k \in \mathbf{Q}} n_{xyk} \log \left(\frac{n_{xyk}\, n_{++k}}{n_{xy+}\, n_{+yk}} \right), \tag{13}$$

where $X, Y \in \mathbf{X}$ and, to prevent notational clutter, their lowercase representations denote their possible states (Kullback, 1959). Let $Q \subseteq \mathbf{X}$ and k be a state of dimension $|Q|$ such that it represents a state for all of the nodes present in Q. Note that in this equation, the subscript "$+$" is used to denote the sum over the respective index, like used in Agresti (2013). For example, n_{x+k} is the count of observations for some particular states x and k, by summing over all the states of node Y. If the p-value for testing the dependency structure encoded in the argument of equation (13) is greater than 0.05, the structure will be deemed not statistically significant and therefore removed from the current graph. However, for discrete, ordinal data the G^2 test does not suffice. Rather, the Jonckheere-Terpstra test (JT) is designed to deal with ordinal data (Jonckheere, 1954), (Terpstra, 1952). This is another non-parametric test that leverages comparisons of the counts of the levels of the current variable, given some other variable. The JT test statistic T is defined as the sum of pairwise comparisons between observations in earlier levels and later levels. Specifically:

$$T = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} T_{ij},$$

where T_{ij} is the count of all pairwise comparisons between values of node Y for levels L_i and L_j of some node X:

$$T_{ij} = \sum_{y_x \in Y_{L_i}} \sum_{y_y \in Y_{L_j}} \mathbb{I}(y_x < y_y),$$

and $\mathbb{I}(x < y)$ is an indicator function:

$$\mathbb{I}(y_x < y_y) = \begin{cases} 1, & \text{if } y_x < y_y, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, this can be understood as splitting dependent variable Y into groups of values according to the levels in independent variable X and counting how many times values for Y are smaller between one level of X and another. In this way, a measure is made about how the values of the dependent variable are distributed, which is subjected to a significance test. In addition, the sequential Monte Carlo permutation method (sMC) is used in order to produce more reliable results for variables that have levels with limited number of observations. The sMC is a more efficient adaptation of the regular Monte Carlo permutation method by adaptively determining when to stop the permutations. Let p_k and p_{k-1} represent the estimated p-values after k and $k-1$ permutations, respectively. The stopping criterion is:

$$|p_k - p_{k-1}| < \epsilon, \tag{14}$$

where ϵ is a predefined threshold for convergence. Furthermore, the observed test statistic T_{obs} is computed, along with an initial estimate of the p-value according to:

$$p_{init} = \frac{\#(T^{(b)}_{\text{perm}} \geq T_{\text{obs}})}{B_{init}},$$

where B_{init} represents the number of calculated permutations T_{perm} and p_{init} is the initial p-value used for equation (14). Iteratively, perform additional permutations according to equation (15), until the stopping criterion from equation (14) is met.

$$p_k = \frac{\#(T^{(b)}_{\text{perm}} \geq T_{\text{obs}})}{k}, \tag{15}$$

With the G^2 or JT independence tests (depending on the data), the adjacency search seen in algorithm 3 can be fully understood. Note that for the pc.stable algorithm the JT test is used. The G^2 test will be resorted to in later examples.

In this algorithm, a pair of adjacent nodes (X_i, X_j) is selected and the adjacency set excluding X_j is considered for further use as long as the number of nodes in the set is greater than ℓ. Consequently, ℓ controls the size of this adjacency set such that for $\ell = 0$, direct dependency between two nodes is tested. However, dependency given some other node(s) is considered for $\ell \geq 1$. Note that as edges are deleted and ℓ grows, the number of nodes that have an adjacency set satisfying $|adj(\mathcal{C}, X_i)\ \{X_j\}| \geq \ell$ shrinks, as this condition becomes harder to fulfill. When the adjacency search is finished and step two commences, unshielded triples (X, Y, Z) are oriented into v-structures if and only if $Y \notin$ sepset(X,Z). Finally, step three orients the remaining edges according to Colombo and Maathuis (2015):

- orient $Y - Z$ into $Y \rightarrow Z$ whenever there is a directed edge $X \rightarrow Y$ such that X and Z are not adjacent (otherwise a new v-structure is created);
- orient $X - Y$ into $X \rightarrow Y$ whenever there is a chain $X \rightarrow Z \rightarrow Y$ (otherwise a directed cycle is created);
- orient $X - Y$ into $X \rightarrow Y$ whenever there are two chains $X - Z \rightarrow Y$ and $X - W \rightarrow Y$ such that Z and W are not adjacent (otherwise a new v-structure or a directed cycle is created).

Algorithm 3 Adjacency search / Step 1 of the PC-algorithm

```
Require: Conditional independence information among all variables in $\mathbf{X}$, and an ordering $\mathbf{V}$ on the variables
 1: Form the complete undirected graph $C$ on the node set $\mathbf{X}$
 2: Let $\ell = -1$;
 3: repeat
 4:     Let $\ell = \ell + 1$;
 5:     repeat
 6:         Select a (new) ordered pair of vertices $(X_i, X_j)$ that are adjacent in $C$ and satisfy $|\text{adj}(C, X_i) \setminus \{X_j\}| \geq \ell$, using order($\mathbf{V}$);
 7:         repeat
 8:             Choose a (new) set $\mathbf{S} \subseteq \text{adj}(C, X_i) \setminus \{X_j\}$ with $|\mathbf{S}| = \ell$, using order($\mathbf{V}$);
 9:             if $X_i$ and $X_j$ are conditionally independent given $\mathbf{S}$ then
10:                 Delete edge $X_i - X_j$ from $C$;
11:                 Let sepset$(X_i, X_j)$ = sepset$(X_j, X_i) = \mathbf{S}$;
12:             end if
13:         until $X_i$ and $X_j$ are no longer adjacent in $C$ or all $\mathbf{S} \subseteq \text{adj}(C, X_i) \setminus \{X_j\}$ with $|\mathbf{S}| = \ell$ have been considered
14:     until all ordered pairs of adjacent vertices $(X_i, X_j)$ in $C$ with $|\text{adj}(C, X_i) \setminus \{X_j\}| \geq \ell$ have been considered
15: until all pairs of adjacent vertices $(X_i, X_j)$ in $C$ satisfy $|\text{adj}(C, X_i) \setminus \{X_j\}| \leq \ell$
16: return $C$, sepset.
```

The third structure learning algorithm is the Max-Min Parents and Children algorithm (MMPC), which is a constraint-based algorithm, but takes a so-called 'local discovery'-approach. This approach involves finding direct parent and child candidate nodes, using similar tests as before, and creating triplets from the candidate set and testing for independence. In particular, the main heuristic behind this algorithm to find candidate edges is the Max-Min heuristic. To properly define this heuristic, we need to introduce a way to find the minimum association between nodes. The association between nodes is measured using a statistical independence test of choice, but the strength of this association given by the p-value will be used to determine the minimum association. To this end, a general definition of this minimum association is presented in definition 2.11.

Definition 2.11. *For some nodes $X_i, X_j \in \mathbf{X}$, the minimum association between nodes X_i and X_j, given a subset of nodes $\mathbf{Z} \subseteq \mathbf{X}$ is given by*

$$MinAssoc(X_i, X_j|\mathbf{Z}) = \min_{\mathbf{S} \subseteq \mathbf{Z}} Assoc(X_i, X_j|\mathbf{S}),$$

where the function Assoc represents the calculation of the strength of the p-value of the statistical independence test.

As shown in definition 2.11, the association strength of the dependency structure between some pair of nodes, is minimized over the possible subsets of remaining nodes. In this way, if the dependence between a pair of nodes is influenced by some subset $\mathbf{S}$, it indicates that this subset is potentially relevant in the dependency structure of the pair. The MMPC algorithm uses these subsets as candidates for parents and children. This set of candidates will be abbreviated as **CPC** (Candidate Parents and Children). Looking at algorithm 4, it can be seen that among that for some iteratively updated **CPC** and target node X_j, the node X_i that maximizes the association is stored. This procedure can be intuitively interpreted as selecting the variable X_i that despite the best efforts to make it independent of X_j, still has the highest such minimum association among all other candidates (Tsamardinos, Aliferis, & Statnikov, 2003). The first phase of the algorithm 4 thus collects all the candidate parents and children by assessing if the association is non-zero (because only potential parents and children will have nonzero association). In phase two, any false positives from phase one are removed by checking if for some $X_i \in \mathbf{CPC}$, statistical independence is returned when conditioning on some subset $\mathbf{S} \subseteq \mathbf{CPC}$. Note that this algorithm does not orient the edges at any point. Therefore, a hybrid approach will be implemented in which a Hill Climbing (HC) algorithm is applied and restricted to the skeleton obtained by the MMPC algorithm. This combination is commonly known as Max-Min Hill-Climbing (MMHC) (Scutari & Denis, 2021).

Algorithm 4 MMPC Algorithm

```
function MMPC(target node T, Data)

% Phase I: Forward
$\mathbf{CPC} \leftarrow \emptyset$
repeat
    $\{F, \text{assoc}F\} \leftarrow \text{MaxMinHeuristic}(T, \mathbf{CPC})$
    if $\text{assoc}F \neq 0$ then
        $\mathbf{CPC} \leftarrow \mathbf{CPC} \cup \{F\}$
    end if
until CPC has not changed

% Phase II: Backward
for all $X \in \mathbf{CPC}$ do
    if $\exists \mathbf{S} \subseteq \mathbf{CPC}$ such that $\text{Ind}(X; T \mid \mathbf{S})$ then
        $\mathbf{CPC} \leftarrow \mathbf{CPC} \setminus \{X\}$
    end if
end for
Return CPC

function MaxMinHeuristic($T, \mathbf{CPC}$)
    $\text{assocF} \leftarrow \max_{X \in V} minAssoc(X; T \mid \mathbf{CPC})$
    $F \leftarrow \arg\max_{X \in V} minAssoc(X; T \mid \mathbf{CPC})$
Return $\{F, \text{assocF}\}$
```

The fourth and final structure learning approach considered in this paper is the information theory-based approach (Margolin et al., 2006), (Scutari & Denis, 2021). This means that the dependency between nodes is quantified using the mutual information test, which is equivalent to the statistic of the G^2 test discussed before (Scutari & Denis, 2021). An important remark to make here is that in contrast to the JT test, the G^2 test does not consider the ordinal nature of random variables in $\mathbf{X}$. For this reason, it strictly focuses on dependencies between particular states of the nodes at hand. Reflecting on equation (13), it can be seen that the entropy is calculated for the target node(s) for all the parent states. Therefore, the resulting p-values can be interpreted as a measure of the significance of the total entropy of the dependency structure at hand. Comparing this to the JT test highlights that the G^2 test reflects a dependency structure across states between some nodes, whereas the JT test reflects monotonically increasing trends that exist in the dependency structure.

Because mutual information will be a recurring topic in this section, let us clearly state its definition. For two discrete random variables X and Y with joint distribution $P(X, Y)$ and marginal distribution $P(X)$ and $P(Y)$, the mutual information is defined as:

$$I(X;Y) \;=\; \sum_{x\in X}\sum_{y\in Y} P(x,y)\,\log\frac{P(x,y)}{P(x)\,P(y)}. \tag{16}$$

Alternatively, it can be defined as

$$I(X_i, X_j) = H(X_i) + H(X_j) - H(X_i, X_j), \tag{17}$$

where the Shannon entropy, $H(X)$, when using the lower case for realizations is:

$$H(X) - \sum_{x} P(x)\log P(x), \quad H(X,Y) = -\sum_{x}\sum_{y} P(x,y)\log P(x,y)$$

.

Most notably, the mutual information is a non-negative metric because neither factor in the sum can be smaller than zero. That is, $I(X;Y) \geq 0$. Secondly, it is a symmetric metric because the multiplication is associative. Additionally, the joint probability is also symmetric by definition because events x and y happening is identical to events y and x happening. Lastly, the mutual information is equal to exactly zero when the variables are completely independent. This is because, under independence, $P(X, Y) = P(X)P(Y)$, which makes the logarithm evaluate to $\log 1 = 0$.

Looking at the ARACNe algorithm displayed in algorithm 5, we can see some striking differences between the previous algorithms. After the mutual information (MI) measure, which is symmetric, is computed for all pairs of nodes possible in $\mathbf{X}$, all of the resulting $\frac{N*(N-1)}{2}$ MI measures for the edges I_{ij} are stored. Subsequently, all of the measures that do not meet the threshold are removed. Lastly, the remaining edges are subjected to the so-called data processing inequality (DPI). With this inequality, the mutual information of a pair (X_i, X_j) is contested by the mutual information of a third pair with either of the nodes of the initial pair: (X_i, X_k) and (X_j, X_k). If the MI score of the initial pair is lower than either of the latter two MI scores, one of these pairs evidently has a stronger relation than the initial pair. Therefore, the edge representing the weaker pair is considered indirect and is removed. As a final remark, it should be mentioned that because of the symmetric nature of the MI measure, no directions can be inferred. Therefore, a hybrid approach similar to MMHC is used to learn the orientation of the edges using the HC algorithm after the undirected structure is learned. The hybridized algorithm will simply be referred to as ARACNe in tables and figures for brevity and more convenient notation.

2.4 Data

Before presenting the other methods used in this research, it is useful to first consider the data used. To this end, the different data sources and their corresponding random variables will be discussed. The data were collected programmatically from various APIs from which data in several granularity was retrieved. In particular, the following sources were tapped into to retrieve data:

- **Google Analytics**. This API contains detailed records of user sessions on the various Custom Decks website

Algorithm 5 ARACNe Algorithm (Adaptive Reconstruction of Accurate Cellular Networks)

```
Require: Discrete data representing $\mathbf{X}$, information threshold $I_0$
Ensure: Inferred network of direct interactions between nodes
 1: for each pair of nodes $(X_i, X_j)$, $i < j$ do
 2:     Compute the mutual information: $I(X_i, X_j) = H(X_i) + H(X_j) - H(X_i, X_j)$,
 3: end for
 4: for each pair of nodes $(X_i, X_j)$, $i < j$ do
 5:     if $I(X_i, X_j) < I_0$ then
 6:         Remove $I(X_i, X_j)$ from the candidate interactions.
 7:     end if
 8: end for
 9: while there exists a triplet $(X_i, X_j, X_k)$, $i \neq j \neq k$ do
10:     if $I(X_i, X_j) \leq \min(I(X_i, X_k), I(X_k, X_j))$ then
11:         Remove $I(X_i, X_j)$ from the candidate interactions.
12:     end if
13: end while
14: Build a graph where:
15: for each remaining pair $(X_i, X_j)$ do
16:     Add an edge $I_{ij}$ to the edge set $\mathbf{I}$, such that $X_i - X_j$ .
17: end for
18: return Final network graph $G(\mathbf{X}, \mathbf{I})$ with direct interactions only.
```

domains. In this data, every row represents a visited page by a unique user, meaning that metrics such as engagement duration measure the time spent on a particular page by a particular user, as well as the amount of pages visited within the session.

- **Meta Analytics**. This API contains measures of the engagement and traction realized by the Custom Decks social media account on Instagram. Metrics such as impressions, reach, likes, comments and referrals to the web shop are measured to monitor the performance of the business' social media activities. The data are measured at a daily level.
- **Custom Decks back-end**. This API contains the sales records, or conversions realized from the web shop. The most important metrics here are turnover on decks and accessories, respectively.
- **KNMI**. This API contains daily records of the weather measurements in the Netherlands. For simplicity, the weather circumstances measured at De Bilt station are used to represent the weather on a given day, as this is the most geographically central station in the Netherlands.

After collecting the data from these sources, they were aggregated to a daily level, if this was not already the case. Although the data are assumed to be present for every single day from every API, this is not necessarily the case. In particular, because sales orders were not received every day, some form of imputation has to be applied to maintain a data frame that has no missing records. In the event of no sales, the turnover variables were set to zero to reflect the fact that no products were sold that day. All other APIs did have daily records and no further imputations were required.

Now, to make the data frame usable for DBN applications, the data were aggregated to rolling sums over some particular windows $T_s \subset \{1, 2, 7, 14, 21, 28\}$, where $t \in T$ stands for the amount of delay in days (rows) that were included in the window. It is important to note that different subsets of $T_s \subset T$ will be implemented. The different subsets and their motivations will be discussed later in section 2.6. Additionally, for two subsequent elements $t_i, t_{i+1} \in T_s$, the data used inside of a rolling window for delay t_{i+1} consist of the $t_{i+1} - t_i$ rows that precede the t_i rows so that no overlap exists in the windows to minimize collinearity. For example, if $T_s = \{7, 14\}$ the resulting windows both use 7 rows, but the window corresponding to $t = 14$ uses the seven rows (days) prior to the first seven rows used by $t = 7$ relative to the current row in the data frame. Furthermore, all of the data in each rolling window was aggregated by averaging the values within the window. This process finalizes the procedure for constructing a DBN node set $\mathbf{X}_{T_s}$.

With regard to the body of data, because not every API logged data starting from the same date, the data used in this research are limited to 366 rows, corresponding to all of the days in the year 2024. Additionally, because only the meteorological data for the Netherlands were retrieved in this research, all web and sales activities were filtered to only consist of activities that originated in the Netherlands as well. Note that the size of the data frame varies depending on the window sizes used. Namely, $t_{max} = \max_{t \in T_s} T_s$ rows are required for the calculation of the windows specified by T_s, resulting in the first $t_{max} - 1$ rows becoming incomplete (because delaying a variable by t_{max} within the first $t_{max} - 1$ rows is practically impossible) and therefore removed from the data frame. The initial data frame consists of a total of 18 variables. However, based on the amount of delays introduced, the number of columns gets multiplied by this amount of delays because all delays $t_i \in T_s$ are applied to all variables.

2.5 Discretization

When using BNs, some forms of data preprocessing might be required to make the data suitable for structure learning. Depending on the resulting data frame, the BN can be a discrete BN, a continuous BN, or a mixed BN. In the latter case, a mixed BN refers to a BN that uses both discrete and continuous variables. However, when continuous variables exist in the data frame, a Gaussian distribution of the data is assumed. However, this assumption is relaxed when using discrete variables because a multinomial distribution is assumed, making discrete BNs accessible in use in many cases. The most well-known discretization methods are interval-based or quantile-based, meaning that the continuous variables are either discretized into bins of equal width or equal frequency respectively. Although these methods are straightforward, the resulting bins do not always properly reflect the nature of the data. For instance, interval-based discretization would be best suited when the data are uniformly distributed. On the other hand, the frequency-based approach is less sensitive to outliers but lacks the sensitivity to capture jumps in the data, meaning that bins might include values within an arbitrary range. Therefore, more advanced discretization methods could prove to be worthwhile for creating more informative labels on which the networks will be trained. One more advanced discretization method to deal with this problem is the Hartemink algorithm from Hartemink (2001). This method leverages pairwise mutual information between random variables and imposes bins on the variables that minimize the entropy between them. This is done by maximizing the sum of the mutual information between each pair of nodes for every node iteratively (Hartemink, 2001). Therefore, this method aims to optimize spread of the bins in order to capture as many meaningful relationships as possible, while guarding itself to overfitting by maximizing the sum of the MI across all pairs, rather than maximizing the MI for a particular pair. This theoretically should benefit the algorithms that use mutual information for their dependence tests, because any true dependencies should be discovered from the new discretization.

It should be noted that, because the Hartemink algorithm aims to optimize performance on mutual information, it does not necessarily focus on the spread of the data of the individual variable. In some cases, this can lead to the algorithm determining bin boundaries that do not appear logical when plotting the values of the random variable along with these boundaries. Of course, this is the result of optimizing MI with respect to all other variables, but it leaves room for a different perspective on discretization that better captures the spread of an individual random variable. The motivation for this is that in the data at hand, there exist many cases where the spread of the data appears clustered. This originates from the fact that many variables are impacted by social media activity. For example, viral posts, or types of posts that tend to regularly cause additional traction for a short period of time, produce levels in engagement metrics that are substantially higher compared to days where no social media traction was created. In this fashion, such engagement metrics can display sudden peaks in their spread with little to no buildup. Therefore, the lack of smooth transitions between levels of values in the data make a case for an alternative discretization method that better reflects this volatile nature.

The pseudocode of the Hartemink algorithm is shown in algorithm 6. After initialization with some fine quantile or interval discretization, bins are merged by iterating over the different variables and including merges that minimize the loss of mutual information. Per variable, this difference can be expressed as the sum of the mutual information difference after a single bin merge for each variable, as shown in equation (18). As is evident from the algorithm, one merge occurs per variable at a time, reducing n by one with each round until the desired number of bins C is obtained. Note that at every merge, some amount of information is lost. All in all, the algorithm represents a heuristic that performs a greedy search. Therefore, only local optima can be found. However, Hartemink (2001) demonstrated that the amount of information retained with only a few bins can be up to 99%. This example shows that the algorithm can produce very meaningful discretizations. Moreover, it is used extensively to great success in gene research (Russell & Norvig, 2022).

$$\Delta I_{i,(a,b)} = \sum_{j\neq i} \left[I(X_i^{\mathcal{B}}; X_j^{\mathcal{B}}) - I(X_i^{\mathcal{B}'}; X_j^{\mathcal{B}}) \right] \tag{18}$$

Algorithm 6 Hartemink Information-Preserving Discretization

```
Require: Data matrix $\mathbf{X}$ with variables $X_1, \dots, X_n$ (continuous), initial fine discretization $\mathcal{B}^{(0)}$ (e.g., many quantile
    bins or unique-value bins) with $r^{(0)}$ bins per variable, target bins $C$
Ensure: Discretization $\mathcal{B}$ with $C$ bins per variable and high total preserved mutual information
 1: $\mathcal{B} \leftarrow \mathcal{B}^{(0)}$
 2: $r \leftarrow r^{(0)}$
 3: Compute counts for all pairs $(X_i, X_j)$ under $\mathcal{B}$ to enable $I(X_i; X_j)$
 4: while $r > C$ do
 5:     for $i = 1, \dots, n$ do
 6:         $(\Delta I_i^\star, (a_i^\star, b_i^\star)) \leftarrow (+\infty, \text{null})$
 7:         for each adjacent pair of bins $(a, b)$ in $X_i$ under $\mathcal{B}$ do
 8:             $\mathcal{B}' \leftarrow$ binning identical to $\mathcal{B}$ except merge $(a, b)$ into one bin for $X_i$
 9:             $\Delta I \leftarrow 0$
10:             for each $j \in \{1, \dots, n\} \setminus \{i\}$ do
11:                 $I_{\text{before}} \leftarrow I\big(X_i^{\mathcal{B}}; X_j^{\mathcal{B}}\big)$
12:                 $I_{\text{after}} \leftarrow I\Big((X_i)^{\mathcal{B}'}; X_j^{\mathcal{B}}\Big)$
13:                 $\Delta I \leftarrow \Delta I + (I_{\text{before}} - I_{\text{after}})$
14:             end for
15:             if $\Delta I < \Delta I_i^\star$ then
16:                 $(\Delta I_i^\star, (a_i^\star, b_i^\star)) \leftarrow (\Delta I, (a, b))$
17:             end if
18:         end for
19:     end for
20:     $r \leftarrow r - 1$
21:     if $r = C$ then break
22:     end if
23: end while
24: return $\mathcal{B}$
```

Complementary to Hartemink discretization, the partitioning around medoids (PAM) clustering method is proposed. A medoid can be described as a central point in a cluster. However, rather than the medoid being an averaged point within the space, a medoid is defined to be an actual observation, and hence a member of the cluster itself. One of the main motivations for using PAM is that in practice, when querying a network, the bins produced by this method tend to be more intuitive in use because they are specific to a single variable, rather than an entire data set. This would make the network query outputs easier to interpret. Moreover, the bins tend to better reflect the patterns observed in the random variable for the very same reason. These properties are especially useful when a user queries the network, because the bins reflect a less abstract discretization.

Now, in terms of formal procedure, the PAM algorithm consists of two phases: the build phase and the swap phase. The build phase is used to find the initial cluster centers that provide good starting points for the second phase (Schubert & Rousseeuw, 2021). The build phase first locates a central medoid that minimizes the total deviation with respect to all other nodes, after which the other $k-1$ initial medoids are determined. This can be seen in lines 1 to 10 in algorithm 7. In the algorithm, the * superscript denotes the optimal results so far in the algorithm. The total deviation is given by

$$TD := \sum_{i=1}^{k} \sum_{x_c \in C_i} d(x_c, m_i), \tag{19}$$

where the distance function in our application is the Euclidean distance function, and $C_i \subset X$ represents one of the k clusters found. After the distance to the first medoid, m_1, is computed and stored in $d_{\text{nearest}}(x_o)$ for each other observation x_o, the second part of algorithm 7 commences. In this part, new medoids are iteratively introduced by the process described in lines 14 to 31. After initializing the current improvement on the distance metric ΔTD and the current best medoid candidate x^*, the algorithm loops over all other non-medoid observations x_o (of which there is $|X|-1$ initially) and computes the distance $d(x_c, x_o)$ between this observation and all remaining observations x_c. If this distance is smaller than the distance between x_c and the current medoid m_1, the difference is negative (as described in line 19 denoted by δ) and the improvement ΔTD is updated. By updating ΔTD if the new total deviation is smaller than the current one and iterating over all observations x_o, the observation that minimizes the total deviation the most is selected as the new medoid. Afterwards, the distance for every observation and its nearest medoid is determined for further iterations in line 30. Note that the number of medoids during this process increases incrementally toward k with each iteration.

Note that this process does not provide the optimal clustering because its initial medoid is the most central one. Although the other medoids are placed optimally with respect to this initial central medoid, the TD value could be improved if we depart from this initial medoid to better represent the structure of the data. For that reason, algorithm 8 was developed to reassign medoids to new observations to minimize TD, but while using the previously assigned medoids from algorithm 7 as informed starting points. The swap algorithm displayed in 8 describes this process.

In this algorithm, an adjustment to equation (19) is made to describe the improvement in the distance between some point x_o and when swapping the currently considered medoid m_i with x_c. The resulting computation is given in equation (20). Furthermore, nearest(o) represents the medoid that is closest to x_o and $d_{\text{second}}(o)$ denotes the distance to the second closest medoid to x_o. Note that in the algorithm, these medoids can differ from the medoid m considered in a particular iteration. Lastly, the function Δ inside of equation (20), shown in equation (21), checks the distance between observations x_o and x_c along with if the medoid under consideration is closest to x_o. Based on the resulting four conditions, either zero, the differences between the non-medoid distances, and between x_o and other medoids are returned. In particular, we can see in equation (21) that not all resulting differences are strictly positive, meaning that equation (20) can be decomposed into positive and negative losses. Consequently, line 16 checks whether the change in total deviation is positive, which would indicate a reduction in performance, and consequently terminates the loop. However, if the computed ΔTD is smaller than the previously optimal ΔTD^*, the medoid is swapped for the observation x_c that instantiated this improvement. With the adjusted set of medoids, the loop starts all over and performs the same steps to find a new potential swap. For a more elaborate demonstration of the workings and variations of these algorithms, we refer to Schubert and Rousseeuw (2021).

Algorithm 7 PAM BUILD: Find Initial Cluster Centers

```
Require: Random variable $X$, number of clusters $k$
Ensure: Initial set of medoids $\{m_1, \ldots, m_k\}$ and total deviation $TD$
 1: $(TD, m_1) \leftarrow (\infty, \text{null})$
 2: for each $x_o \in X$ do
 3:     $TD_x \leftarrow 0$
 4:     for each $x_c \in X \setminus \{x_o\}$ do
 5:         $TD_x \leftarrow TD_x + d(x_c, x_o)$
 6:     end for
 7:     if $TD_x < TD$ then
 8:         $(TD, m_1) \leftarrow (TD_x, x_o)$
 9:     end if
10: end for
11: for each $x_o \in X \setminus \{m_1\}$ do
12:     $d_{\text{nearest}}(x_o) \leftarrow d(x_o, m_1)$
13: end for
14: for $i = 1, \ldots, k-1$ do
15:     $(\Delta TD, x^*) \leftarrow (0, \text{null})$
16:     for each $x_o \in X \setminus \{m_1, \ldots, m_i\}$ do
17:         $\Delta TD \leftarrow 0$
18:         for each $x_c \in X \setminus \{m_1, \ldots, m_i, x_o\}$ do
19:             $\delta \leftarrow d(x_c, x_o) - d_{\text{nearest}}(x_c)$
20:             if $\delta < 0$ then
21:                 $\Delta TD \leftarrow \Delta TD + \delta$
22:             end if
23:         end for
24:         if $\Delta TD < (\Delta TD)^*$ then
25:             $(\Delta TD^*, x^*) \leftarrow (\Delta TD, x_o)$
26:         end if
27:     end for
28:     $(TD, m_{i+1}) \leftarrow (TD + (\Delta TD)^*, x^*)$
29:     for each $x_o \in X \setminus \{m_1, \ldots, m_{i+1}\}$ do
30:         $d_{\text{nearest}}(x_o) \leftarrow \min(d_{\text{nearest}}(x_o), d(x_o, m_{i+1}))$
31:     end for
32: end for
33: return $TD, \{m_1, \ldots, m_k\}$
```

Algorithm 8 PAM SWAP: Iterative Improvement

```
Require: random variable $X$, set of medoids $\{m_1, \ldots, m_k\}$
Ensure: Optimized set of medoids $\{m_1, \ldots, m_k\}$ and total deviation $TD$
 1: foreach $x_o \in X$ do
 2:     Compute nearest$(o)$, $d_{\text{nearest}}(o)$, $d_{\text{second}}(o)$
 3: repeat
 4:     $(\Delta TD^*, m^*, x^*) \leftarrow (0, \text{null}, \text{null})$
 5:     for each medoid $m \in \{m_1, \ldots, m_k\}$ do
 6:         for each non-medoid $x_c \in X \setminus \{m_1, \ldots, m_k\}$ do
 7:             $\Delta TD \leftarrow 0$
 8:             for each $x_o \in X \setminus \{m_1, \ldots, m_k\}$ do
 9:                 $\Delta TD \leftarrow \Delta TD + \Delta(x_o, m, x_c)$
10:             end for
11:             if $\Delta TD < \Delta TD^*$ then
12:                 $(\Delta TD^*, m^*, x^*) \leftarrow (\Delta TD, m, x_c)$
13:             end if
14:         end for
15:     end for
16:     if $\Delta TD^* \geq 0$ then
17:         break loop
18:     end if
19:     Swap roles of medoid $m^*$ and non-medoid $x^*$
20:     for each $x_o \in X$ do
21:         Update nearest$(o)$, $d_{\text{nearest}}(o)$, $d_{\text{second}}(o)$
22:     end for
23:     $TD \leftarrow TD + \Delta TD^*$
24: until convergence
25: return $TD, \{m_1, \ldots, m_k\}$
```

$$\Delta TD = \sum_{x_o} \Delta(x_o, m_i, x_c) \tag{20}$$

$$\Delta(x_o, m_i, x_c) = \begin{cases} 0 & \text{if } d(x_o, x_c) \geq d_n(o) \text{ and nearest}(o) \neq i \quad \text{(a)}, \\ d(x_o, x_c) - d_n(o) & \text{if } d(x_o, x_c) < d_s(o) \text{ and nearest}(o) = i \quad \text{(b1)}, \\ d_s(o) - d_n(o) & \text{if } d(x_o, x_c) \geq d_s(o) \text{ and nearest}(o) = i \quad \text{(b2)}, \\ d(x_o, x_c) - d_n(o) & \text{if } d(x_o, x_c) < d_n(o) \text{ and nearest}(o) \neq i \quad \text{(c)}. \end{cases} \tag{21}$$

With this formulation of the PAM algorithm, each accepted swap constitutes an improvement of the cost function. This means that the algorithm is a greedy method. Moreover, the number of possible medoid sets is finite, i.e. $\binom{N}{k}$, meaning that the algorithm cannot search indefinitely. By definition, when no proposed swaps would improve the cost function, the algorithm terminates. This means that the PAM algorithm discovered a local optimum. Although exceptions exist by construction, the algorithm often finds a good, if not the optimal result (Tiwari et al., 2023).

The four different discretization methods presented in this paper are displayed in figure 1. In this example, all plots were created with three different bins. Note the similarity between the quantile plot 1b and Hartemink plot 1c. The bin representing the smallest observations better captures the range of these small value observations in the case of plot 1c. This implies that these low values yield high mutual information values in combination with some other variables. However, the third bin that represents the highest values contains all values greater than 23, which covers a relatively large part of the possible range of values. This means that when querying the network, it is practically impossible to distinguish between these different high and moderate values. In contrast, the medoid (PAM) discretization provides a more mixed approach. In figure 1d it can be seen that the algorithm also recognizes the need for a narrow bin to capture the low-value observations, but also provides a bin for moderately high values. The latter results in a discretization that might make more sense for a user, similar to an interval discretization, but provides a more nuanced clustering that results in a more informative discretization.

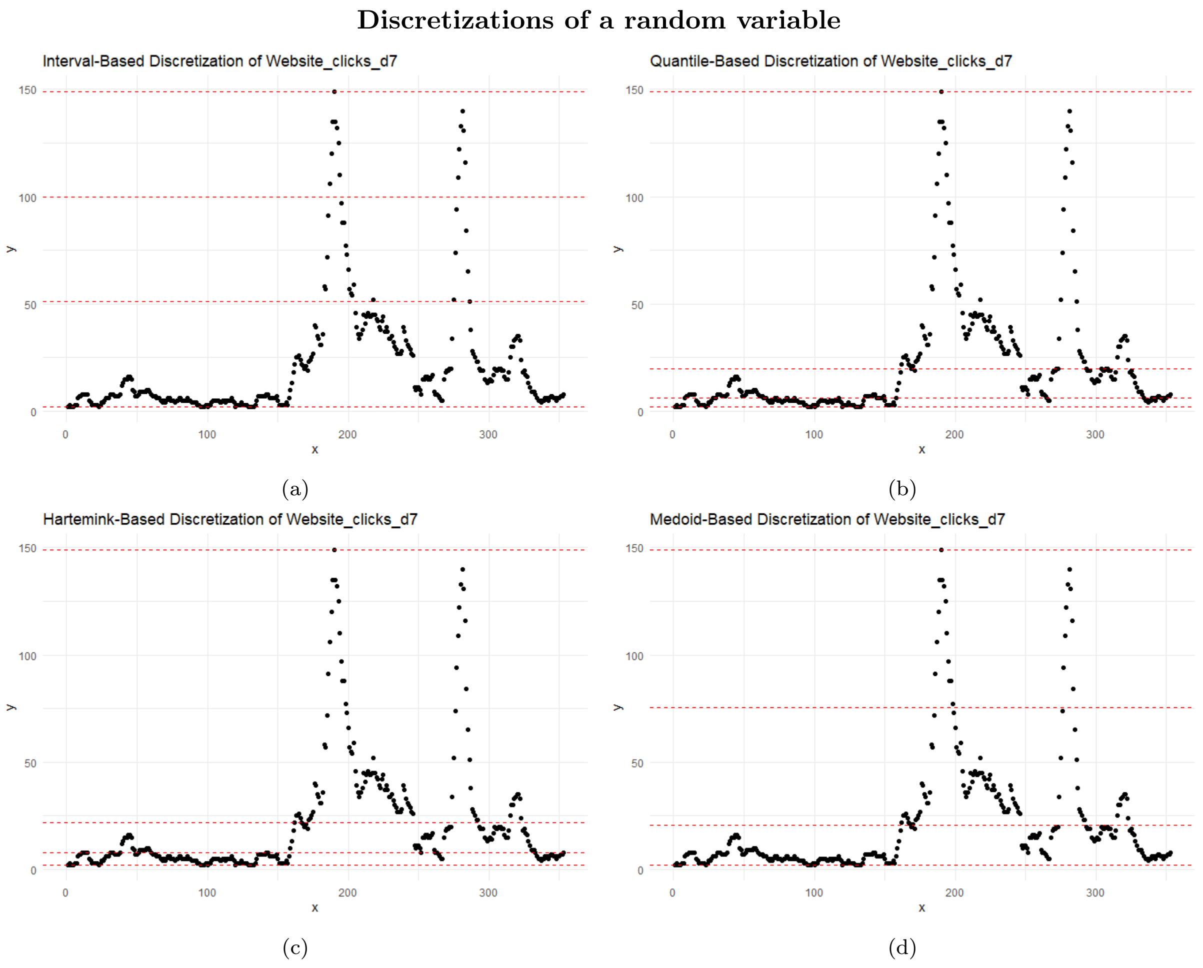


Figure 1: Four plots with different discretizations of the same variable. The horizontal axis contains the indices of the aggregated observations and the vertical axis represents the attained value of the random variable. The interval discretization provides the most straightforward bins based on the observed range values the variable attains. The quantile discretization has the same amount of observations in each bin and the Hartemink discretization optimizes pairwise information between all pairs of variables.

2.6 Delay analysis

In order to make informed decisions about what delays to impose on the random variables $\mathbf{X}$, a preliminary analysis will be performed. Given that we will be introducing delays into the data, the nature of the problem is similar to that of a time-series analysis. For this reason, the normalized cross-correlations are computed between several aggregated delays (i.e. before discretization) so that patterns such as cyclical behavior can be discerned. In case any meaningful patterns arise, these can be leveraged to construct DBN data sets $\mathbf{X}_{T_s}$ with more robust associations while not having to exhaust all possible delay configurations to discover the more informative DBN data sets. Consider random variables $X, Y \in \mathbf{X}$ and let us for now denote the i-th observation of a random variable as $X(i)$. Then, for some delay of z rows, the normalized cross-correlation (NCC) is defined as:

$$\rho(X,Y,k,M) = \frac{\sum_{i=k-M+1}^{k} X(i)Y(i-z)}{\sqrt{\sum_{i=k-M+1}^{k} X^2(i)}\sqrt{\sum_{i=k-M+1}^{k} Y^2(i)}}, \tag{22}$$

where M and k are hyperparameters that control the number of rows used to compute the cross-correlation and $k \leq N$ is the index of the final row to be used respectively. With hyperparameter M, one is able to set the bandwidth of the NCC. That is, larger M results in more rows of the random variables are used, whereas smaller M do the converse. Controlling this parameter can be useful because some patterns might occur within a limited amount of delay. For example, choosing M close to k, the cross-correlation becomes smoother, which can cause effects to become less salient. Hyperparameter k provides a less interesting role, but can be useful to use in combination with M to deal with phenomena such as a burn-in phase. As can be seen in equation (22), a total of N cross-correlations are summed so that a total correlation within bandwidth M is computed. This number is subsequently normalized by the product of magnitudes of the random variables. This formulation is equivalent to a normalized version of a discrete convolution.

An important side note here is that this function assumes the random variables to have a stationary distribution, meaning that they do not depend on time. The stationarity of a random variable was tested using the augmented Dickey-Fuller test (Fuller, 1996), after the mean of the random variable was subtracted from its values. In short, the test consists of setting up an autoregressive model and testing if the coefficients of the autoregressive terms are significantly different from the null hypothesis that the polynomial has a unit root. In other words, if the p-value when testing the coefficient of the first-order lag of the random variable is not below 0.05, then the random variable is strongly correlated to itself and is therefore not stationary (Fuller, 1996). If a random variable was found to be non-stationary, it was consecutively subjected to differencing and detrending.

After making all variables stationary, the NCC values between all pairs of random variables were computed for different configurations of delays. These NCC values were then averaged for each delay configuration so that the mean NCC for this configuration was returned. With this method, a high-level insight was created on the relatedness of variables when subjected to specific delays.

2.7 Validation methods

One of the main goals of this research is to find methods and configurations in preprocessing and structure learning that provide stronger models for the novel application of BNs to a broad collection of business data. Because many different models, delay configurations, and discretizations will be tested, several validation methods are employed to assess model performance and robustness. Starting with the most basic metrics on model complexity, we will compare the relative network density (in terms of edges), average node degree, maximum path lengths of models, and numbers of nodes and edges. These metrics have straightforward definitions shown below.

$$\text{relativeDensity}(G) = \frac{|\mathbf{I}|}{n(n-1)}, \tag{23}$$

where $|\mathbf{I}|$ is the number of edges in the network, n the number of nodes in the network and $n(n-1)$ maximum possible number of directed edges in a fully connected directed graph.

$$\text{averageDegree}(G) = \frac{\sum_{i=1}^{n} \text{degree}(X_i)}{n}, \tag{24}$$

where $\text{degree}(X_i)$ is the degree (number of incoming and outgoing edges) of node X_i.

$$\text{maxPathLength}(G) = \max\left(\text{length of all directed paths}\right). \tag{25}$$

Additionally, more involved methods are used to assess the complexity of the DBNs. Namely, the network entropy as well as Kullback-Leibler (KL) distances are used. The entropy metric can be used to assess the complexity of a network. Additionally, the KL values represent the relative entropy between two models (Scutari, 2024). The entropy is given by equation (26). This describes the Shannon entropy for a probability distribution P across the entire node set $\mathbf{X}$. With the KL distance measuring the relative entropy, we can see in equation (27) that this measure is asymmetric. That is, $KL(P \parallel Q) \neq KL(Q \parallel P)$, unless the distributions are identical. Effectively, equation (27) describes how well distribution P is explained under distribution Q (Koller & Friedman, 2009). For example, if distributions P and Q are highly similar, $\log(\frac{P(x)}{Q(x)})$ will be closer to 0, meaning that their KL distance will be smaller. These values are then weighted by values of $P(x)$ so that (dis)similarities are weighted by the probability of the respective state. The asymmetry of this measure can be clearly illustrated with the following example. Let P represent some distribution of a model that is overfitting to some extent compared to distribution Q, which is smoother. In that case, $P(x)\frac{P(x)}{Q(x)})$ can produce higher values when $P(x)$ is large, while $Q(x)$ is small. This can happen when overfitting leads to P assigning large probabilities to states that are less likely under distribution Q. In contrast, $Q(x)\frac{Q(x)}{P(x)})$ will be much smaller in this scenario. Therefore, when P is comparatively overfitting to Q, this generally implies $KL(Q \parallel P) < KL(P \parallel Q)$. Therefore, this metric provides insight into the similarity of networks' distributions, as well as which distribution provides smoother estimations.

$$H(P) = \mathbb{E}_{P(\mathbf{X})}\big(-\log P(\mathbf{X})\big) = -\sum_{\mathrm{Val}(\mathbf{X})} P(x)\log P(x)\,dx. \tag{26}$$

$$\mathrm{KL}(P \parallel Q) = \mathbb{E}_{P(\mathbf{X})}\left(-\log\frac{P(\mathbf{X})}{Q(\mathbf{X})}\right) = -\sum_{\mathrm{Val}(\mathbf{X})} P(x)\log\frac{P(x)}{Q(x)}dx. \tag{27}$$

The use of the KL metric requires knowledge of the graph's parameters. This is not done during structure learning, so this needs to be done in a separate step. Learning the parameters can be done in a variety of ways for discrete networks, with the most common being a frequentist MLE approach that simply uses the observed relative frequencies, or the Bayesian approach where a uniform Dirichlet prior is assumed and the posterior distribution is learned. Because some bins could be empty under some parent configurations, the Bayesian approach was used to fit the parameters so that no attainable value for a node would be assigned a zero probability. This would reflect the fact that even though the event has not been observed yet under the given parent configuration, it is not impossible that such events would occur at some point in the future. Hence, by using a Bayesian prior, the belief that unobserved events are possible is captured.

Now, by assuming an imaginary sample size of 1 across all dependencies, $\alpha_{ijk} = 1$. The values of the conditional probability tables are then filled with the posterior means of the multinomial probabilities:

$$\hat{\theta}_{ijk} = \mathbb{E}[P(X_i = k \mid \mathrm{Pa}(X_i) = j)] = \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \sum_{k=1}^{r}\alpha_{ijk}},$$

In addition to the BIC, some metrics that quantify the predictive and probabilistic performance will be used to assess model performance. Predictive power is very relevant for the application at hand, because it will determine the extent to which business decisions can be supported by the data and the network. To this end, the predictive power of a model will be restricted in scope to discovered edges. This means that nodes having no edges and thus lacking a dependency structure are excluded from the predictive performance assessment. Now, because not all models will be providing the same number of edges, a balance must be struck between the number of variables that can reliably be predicted and the strength of this prediction. Therefore, the more elementary metrics on nodes and edges will contextualize the models' performances on the prediction metrics. For example, if a network has very good predictive metric values, but at the cost of a very low number of edges, it indicates that the network might not be suitable after all. The metrics that will be considered are the Brier score, the precision-recall (pr) curve and its area under the curve (AUC), and the predictive accuracy. The Brier score represents the probabilistic accuracy of a model. In particular, it can be defined given a proper CPT that represents the probability distribution of a dependency structure. The normalized Brier score (BS) is then given by:

$$BS = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{r}\left(C_{ik} - p_{ik}\right)^2, \tag{28}$$

where N is the number of observations, r is the number of classes present in the child node under consideration, C_{ik} is a binary indicator that represents the class returned by the model, and p_{ik} is the probability that this class occurs as mentioned in the CPT (Kruppa et al., 2014). In this manner, classes that come with high probabilities given the information from the parent nodes are rewarded with low scores. In contrast, classes that occur while low probabilities were assigned respectively, will result in bigger differences and hence increase the score. In other words, it is a probabilistic measure of model accuracy, where lower scores are preferred. The Brier scores for all the child nodes are aggregated such that each model has an averaged Brier score.

In addition to the Brier score, the precision-recall (pr) curves are computed to assess how well the states of child nodes can be predicted given their parent states. For completeness' sake, let us briefly address how the curve is defined. First of all, the precision is given by

$$PR = \frac{TP}{TP + FP}, \tag{29}$$

where TP and FP represent true and false positives, respectively. Secondly, recall is defined as

$$RC = \frac{TP}{TP + FN}, \tag{30}$$

where FN represents the number of false negatives. In short, the precision quantifies how many of the positive cases were correct, whereas recall quantifies how many of the factually positive cases were correctly labeled positive by the model. Typically, there is a trade-off between these two quantities when trying to optimize one or the other, and which quantity is preferred depends on the application at hand. For instance, high recall is preferred in medical contexts to prevent false negatives, which would result in care not being administered to people actually needing it. This trade-off can be plotted in a precision-recall curve, to see which levels of recall are attained at which levels of precision. However, for our application, a more holistic approach is used, where the area under the curve (AUC) for this precision-recall curve is computed. Under this interpretation, the value of 0.5 implies the model would be as good as random guessing. In contrast, a number closer to 1 represents a perfect classifier. In the upcoming model comparison results, these AUC values are averaged across the total number of predicted variables, i.e., all child nodes. These averaged AUC values will be considered along with the number of (child) nodes in the network to contextualize the AUC performance. The AUC value for a particular variable will be defined by the weighted average of the AUC values for each state, where the weights assigned equal the relative frequency of the state within the data used. In this way, frequently occurring states have a bigger influence on the total AUC value for the respective variable.

Lastly, bootstrapping is used to investigate the strength of the edges. The bootstrapping process is performed by sampling values with replacement, as well as permuting observations within random variables to further test the robustness of the associations indicated by the edges that are learned in the structure learning process. As a result, some associations might not appear consistently in models for all bootstrapped samples of $\mathbf{X}$. After the bootstrapping process, model averaging can be used to represent the relative frequency the edge was included in the models that were learned from the bootstrapped samples. These relative frequencies will be referred to as the edge strength. Similarly, the direction of the edge can be inferred by monitoring the direction of included edges across the different models learned during the bootstrapping process. The resulting edges will all have a certain strength and (ambiguous) direction, which can be subjected to a particular strength threshold. Edges will be accepted into the BN if the strength of the edge is greater than this threshold. In this way, an averaged BN is retrieved that only includes edges that remain after the edge strength check is performed, after which the metrics discussed above are computed. The threshold used in this research was 0.75.

To have a sanity check for the structure learning process, the process was applied to the Hailfinder benchmark data set with a known network, taken from Abramson, Brown, Edwards, Murphy, and Winkler (1996). This data set is a discrete synthetic data set that describes a variety of measurements that are used to forecast summer hail in northeastern Colorado. The results found on this data set provide a valuable baseline reference of the power of the structure learning processes and bagging processes used, because the learned structures can be checked against a known network structure. That is, applying our structure learning process to this data set should reveal how well the process manages to find the true network, which is an indicator of how effective the structure learning process is by measuring the learned networks' sensitivity and specificity values. Moreover, it provides some initial information about the appropriateness of the configurations of the structure learning inputs (thresholds, alpha values etc.) for the algorithms used. To this end, the bootstrapping and model averaging methods were applied to this discrete data set and will be displayed in the results section. The same penalty levels and alpha values were used. Only the inclusion threshold after model averaging was reduced from 0.75 to 0.5 to have more stable results across penalty levels. This produces more stable results because with different penalty levels certain arcs may compete to different degrees, resulting in lower relative frequencies, which results in a lower numbers of arcs passing the 0.75 threshold. The higher threshold was still used for the Custom Decks data set to ensure that only strongly supported associations were included.

2.8 Experimental setup

The initial data frame that represents $\mathbf{X}$ consists of 366 rows and 16 columns, where every row represents a single day. Because we have several sources for our data, we will present their origin for each column, as displayed in table 1. This will be handy in the results section of this paper because one of the points of interest is the existence of associations between nodes from different sources. The data were obtained from the official APIs from Meta, Google Analytics and the KNMI and were anonymous by default. The Custom Decks data were retrieved from the business' privately hosted API. Any personal data obtained from this resource were discarded before preprocessing the data.

In the preprocessing phase, only a handful of operations were performed. Most notably, the random variables were made stationary in case they were not already so that the data were usable for the ADF-test (Fuller, 1996). Stationarizing a variable would by manipulating with one method and checking for stationarity before proceeding with another operation. The selected methods, in order, were first-order differencing, detrending and log-transforming. A more detailed explanation of the ADF test and its result for the data set is provided in Rook (2025). The resulting data frame was maintained for the structure learning algorithms such that distributions evolving across the year because of increasing amounts of customers would not confound the results.

In order to compare the different types of discretization, the data frame was subjected to a total of four types of discretization. Namely, these are interval, quantile, Hartemink and medoid (the aforementioned PAM) clustering. This means that four different data frames, one for each discretization method, were used for structure learning.

Secondly, in order to keep track of what delay a variable was subjected to, all variable names are suffixed as '_dt', where t represents the delay. As mentioned in section 2.4, the rows used to compute the aggregated values in a window referred to by '_dt_{i+1}' are the $t_{i+1} - t_i$ number of preceding rows used in the t_i window so that there is no overlap in the rows that were used in a window.

Using this suffix naming convention, blacklists were dynamically constructed to create blacklists necessary for purely causal DBNs and DBNs that do allow co-occurrence between contemporary variables. However, some nodes serve special roles in the networks and were blacklisted manually because they represent external influences or final results in a chain of events. In particular, the KNMI variables are external variables that can only influence other variables, rather than be influenced by the business operations or customer behavior. If we denote a blacklist as B, we can define different partial blacklists that cover different undesired edges. As such, let us store the edges regarding the external variables in $B_{\text{external}} \subset B$. Similarly, the Custom Decks variables that cover the turnover on decks or other items are considered the endpoints of a customer conversion journey and are therefore only allowed to be non-parent nodes. Therefore all edges originating from these variables are blacklisted in $B_{\text{child}} \subset B$.

To this end, the following rules were implemented for dynamic blacklisting, in which we will refer to variable name suffixes '_dt' as just t for more convenient notation. Note that the blacklist is constructed as a data frame with columns "from" and "to", where each column represents the node the edge originates from and points towards respectively. Additionally, superscripts t_i^f and t_j^t will be used to denote if a delay t_i is assigned to a "from"-node or a "to"-node respectively, a pair (t_i^f, t_j^t) represents a row in B, for some $t_i, t_j \in T_s$.

1. $\forall (t_i^f, t_j^t) \in B, \quad t_i^f < t_j^t$
2. $\forall (t_i^f, t_j^t) \in B, \quad t_j^t \neq t_{min}, \quad t_{min} = \min_{t \in T_s} T_s$
3. $t_i^f = t_j^t \Rightarrow (t_i^f, t_j^t) \in B$

Note that the enumeration above lists the conditions for edges that are not allowed into the model and that rule 3 is only implemented for strictly causal DBNs, B_{causal}. Otherwise, only rules one and two are used when allowing edges between contemporary nodes and we refer to this blacklist as B_{co}. The first rule puts all edges in the blacklist that point backward in time (i.e. the to-node has a greater delay than the from-node). The second rule states that pairs in the blacklist cannot have a "to"-node that has the minimum delay. This effectively blacklists all edges between variables that do not point towards the most contemporary nodes. Lastly, the third rule is included in DBN blacklists because it blocks all edges that suggest co-occurrence in contemporary nodes.

The preliminary analysis consists of an exploration of the cross-correlations between aggregated values of delayed variables. This exploration resulted in the incorporation of single day delays or weekly delays. To investigate the robustness of associations between these delay windows in the models, two data frames were created that included most of the aforementioned delays $T_d = \{1, 2, 7, 14\}$ or only included the weekly delays $T_w = \{7, 14, 28\}$. This was motivated by the fact that upon exploration of the data, weekly cyclical trends were observed that could be complementary to the shorter-term trends. The two resulting data sets will be represented in subscripts by $\mathbf{X}_{T_d}$ and $\mathbf{X}_{T_w}$ respectively. Additionally, the type of discretization applied to this data set will be denoted, along the previously discussed subscript, in the superscript as $\mathbf{X}^{\text{interval}}, \mathbf{X}^{\text{quantile}}, \mathbf{X}^{\text{Hartemink}}, \mathbf{X}^{\text{PAM}}$ for the respective type of discretization. The validation methods will be employed for all combinations the algorithms (HC, pc.stable, MMHC, ARACNe-HC), discretizations, delay sets, resulting in 32 configurations. These configurations will also be subjected to the different blacklist constraints in order to investigate the cost of having strictly causal networks for a total of 64 configurations. For ease of notation, these configurations will be denoted as "delay set - discretization - blacklist" in the tables in appendices A and B. For example, one configuration is represented as $\mathbf{X}_{T_s}$-PAM-B_{co}, for some delay set T_s, PAM discretization and where B_{co} stands for the co-occurrence-allowing blacklist.

It is assumed that such a network comes at some cost in terms of performance metrics, because co-occurring events appeared to have stronger correlations than strictly causal events.

Table 1: Variables retrieved from the various APIs

Meta	**Custom Decks**	**Google Analytics**	**KNMI**
FollowersPerDay	Turnover_decks	ActiveUsers	Temperature
Reach	Turnover_other	newUsers	Precipitation_sum
Impressions		userEngagementDuration	
Accounts_engaged		screenPageViews	
Profile_views			
Website_clicks			
like_count			
comments_count			

Furthermore, the structure learning processes for each algorithm will be subjected to different penalties. This was done to control the sparsity of the networks obtained, which will be used to check for overfitting and robustness of the networks provided by the algorithms. To this end, the parameter factor that penalizes the BIC score from equation (10) is of the form $p_a \log(N)/2$, where $p_a \in \{2^{-7}, 2^{-6}, ..., 2^2\}$ has exponential increments and N represents the sample size, as before. This measure introduces varying degrees of penalization for the inclusion of more terms, resulting in denser or sparser graphs. However, because structure learning for pc.stable does not depend on the BIC score, but rather on the statistical significance measured by including an edge and heuristics, the α threshold for the p-value is instead scaled by $\frac{2^{-7}}{p_a}$. In this fashion, increasing the penalty p_a with each iteration shrinks this quotient, lowering the required p-value by a factor of 2 as a result. These penalties will be applied to all of the discretization methods and structure learning algorithm configurations to check their respective performances. For brevity, the subscript used to denote penalty value p_a will represent the power of 2 used to scale the BIC penalty term or the p-value of the statistical significance tests in constraint-based parts of the algorithms. For example, $p_0 = 1$ and $p_{-3} = 0.125$

With the different types of discretizations, delay sets and blacklists set up, models were trained using the ARACNe-HC, HC, MMHC and pc.stable algorithms. These were subjected to bootstrapping with replacement for a total of 500 bootstrapped data sets. The resulting averaged models and their edges were subsequently subjected to a model averaging threshold of 0.75, meaning that the edges had to appear in at least 75% of the models in order to be included in the averaged model. The averaged models for each of these algorithms were computed for the various degrees of penalization to investigate what degree of penalization is desirable. This desirability of a model is quantified using a variety of metrics being the Brier score, the precision recall area under the curve (pr-AUC), BIC and model entropy and assessing these in context of the more elementary network metrics such as the amount of connected nodes and edges in the network. By combining these metrics, the quality and performance of a model can be assessed in context of the corresponding complexity of the model. In combination with the different levels of penalization, these comparisons provide information on which level of penalization provides a suitable level of performance and complexity. For example, a network that has near perfect predictive qualities inside of its dependency structures might only be able to do so because of a small set of edges that happen to have very strong predictive qualities. Such an overly simple network provides too little information to be desirable in practice. In contrast, models that are too dense can be too hard to interpret or include weaker interactions that might obscure the truly impactful dependencies. Therefore, assessing model performance in context of its complexity will allow for choosing sensible models. The results of the validation method with the Hailfinder network shown in the preliminary analysis part of the results section will inform us on what number of edges might be considered too dense or sparse.

The predictive performance of the models will be assessed with the aforementioned probabilistic Brier score and pr-AUC metrics. Using an 80/20 training and test split on the data without shuffling to maintain the temporal structure of the data, the tail 20% of the data set is predicted using the averaged model that was fitted based on the first 80% of records. Using this type of train-test split provides an honest way of assessing the model's performance on time-series data. In addition, the BIC and entropy metrics will be used to assess quality of fit and model complexity, respectively. In combination with the probabilistic metrics, these metrics will provide a more complete view of the parameter efficiency of the model.

Lastly, different amounts of bins will be considered when applying the discretization methods. The networks learned when using different amounts of bins within an experimental setup can be compared on predictive performance by using, among other metrics, the Brier score. Under the assumption of uniform distributions for the variables in our data set, the expected value of the Brier score of a variable can be described as

$$\mathbb{E}[\mathrm{BS}] = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(C_{ik} - \frac{1}{K}\right)^2$$

Afterwards, this can be split into two different cases:

- When $k = y$:
$$\left(1 - \frac{1}{K}\right)^2 = \frac{(K-1)^2}{K^2}$$

- For $K-1$ times where $k \neq y$:
$$\left(0 - \frac{1}{K}\right)^2 = \frac{1}{K^2}$$

Hence, when replacing the inner sum using these factors, one gets

$$\sum_{k=1}^{K}\left(C_{ik} - \frac{1}{K}\right)^2 = \frac{(K-1)^2}{K^2} + (K-1)\cdot\frac{1}{K^2} = \frac{(K-1)^2 + (K-1)}{K^2} = \frac{(K-1)(K-1+1)}{K^2} = \frac{(K-1)K}{K^2} = \frac{K-1}{K}$$

Note that because the latter result is independent of i, the expected value is simply $\frac{K-1}{K}$. Now that the expected value of the Brier score is known, the expected difference in Brier score can be computed when increasing the amount of bins, as demonstrated below.

$$\Delta\text{BS} = \mathbb{E}[\text{BS}_{K+1}] - \mathbb{E}[\text{BS}_K] = \left(\frac{K}{K+1}\right) - \left(\frac{K-1}{K}\right) = \frac{1}{K(K+1)}$$

In case different numbers of bins are used within some data set, the expected increase in average Brier score for a data set can be computed as the weighted average of the differences between the number of bins between the respective variables. In the research at hand, three to at most five bins will be used, which means that increasing the number of bins for some variables beyond three leads to an increase in the expected Brier score as quantified in equation (31), where n_4 and n_5 represent the number of variables with four or five bins, respectively

$$\Delta\text{BS} = \frac{1}{N}\left(n_4\left(\frac{3}{4} - \frac{2}{3}\right) + n_5\left(\frac{4}{5} - \frac{2}{3}\right)\right) \tag{31}$$

To determine the number of clusters that best represents the data, the PAM algorithm was iteratively applied to see what number of clusters would minimize the overall distance within a cluster. More specifically, the silhouette of a cluster is determined to assess how many clusters are appropriate for a variable. This is done by defining the average distance for each point relative to each other point it shares its assigned cluster with; and the average distance to the points in the closest cluster. Respectively, these average distances are mathematically defined as

- **Average intra-cluster distance** $a(i)$:
$$a(i) = \frac{1}{|C_i| - 1}\sum_{\substack{x_j \in C_i \\ j \neq i}} d(x_i, x_j)$$
This is the average distance from x_i to all other points in the same cluster.

- **Lowest average inter-cluster distance** $b(i)$:
$$b(i) = \min_{C \neq C_i}\left(\frac{1}{|C|}\sum_{x_j \in C} d(x_i, x_j)\right)$$
This is the minimum average distance from x_i to all points in any other cluster $C \neq C_i$.

Using these metrics and a specified range of allowed number of bins, the number of bins k^* that maximizes the average silhouette is then selected as the number of bins that best clusters the data, where the silhouette is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{where } -1 \leq s(i) \leq 1$$

and the optimal number of clusters equals

$$k^* = \arg\max_k \bar{s}(k), \tag{32}$$

where $\bar{s}$ denotes the average silhouette across all observations of the variable.

3 Results

3.1 Preliminary analysis

In order to have a baseline for the performance of the structure learning process and different structure learning algorithms, the same methods were applied to the Hailfinder data set used in Abramson et al. (1996). The results are shown in figure 2. In the plots, the blue horizontal line represents the 66 true arcs of the Hailfinder network. The red line represents the correct arcs learned at each penalty level. The dashed black line represents the sum of the true and false positives and hence reflects the number of arcs of the learned network. Therefore, the number of correct arcs can be weighed against the number of incorrect arcs, as well as missing edges.

The results in figure 2 show different behaviors for different algorithms. The ARACNe algorithm seems to find the majority of arcs, while having a limited number of false positives to the extent that the learned network's size is similar to the true network. This shows that the algorithm displayed a low degree of overfitting, but at best found around 50 out of the 66 true arcs. Interestingly, the number of arcs in the learned network seems to decrease slightly with increasing penalty. Given the fact that the alpha of 0.50 was used for the smallest penalty value, it shows that the ARACNe algorithm and its independence tests would not learn denser networks when reducing the specificity of the tests. The behavior of the HC algorithm is markedly different. That is, the number of arcs clearly decreases with increasing penalty values, while the number of correct arcs fluctuates between different ranges of penalties. It appears that with denser networks, more correct arcs are included, whereas with the learned networks becoming sparser, the number of correct arcs initially drops for the intermediate penalty values. However, at the penalty value of p_0, the line attains a value of 54 of correct edges, with a network size comparable to the true network. This fact that not all correct edges would be identified was expected because the paper by Abramson et al. (1996) mentions that some edges were manually manipulated in the true network for reasons such as model tractability and realism. Given that this instance reflects the case where HC was used with a standard BIC score, it should be expected that this solution is the best solution, given the fact that the standard BIC score tends to the BDe score. Although the result with a penalty value of p_1 contains fewer correct arcs, this stricter version of the HC algorithm creates a network that has a better specificity for arc detection. This suggests a trade-off between number of correct arcs and specificity for increasing penalty values under the HC algorithm. Lastly, we have the MMHC and pc.stable algorithms that display similar behavior. Both find too few arcs. Moreover, the most dense networks for these algorithms at the lower penalty levels still have relatively poor specificity. In contrast, it appears that the algorithms are very robust with respect to the number of correct edges. Nevertheless, these algorithms seem to be a poor fit given the fact that an alpha threshold of 0.5 was used and the algorithms would not find networks even of appropriate density nor sensitivity. All in all, with these results serving as a baseline, it can be argued that the networks learned from the HC algorithm under a penalty value of p_0 could be viewed as the most appropriate network and the networks of the other algorithms can be compared against this network. This network will be termed the 'reference HC network'.

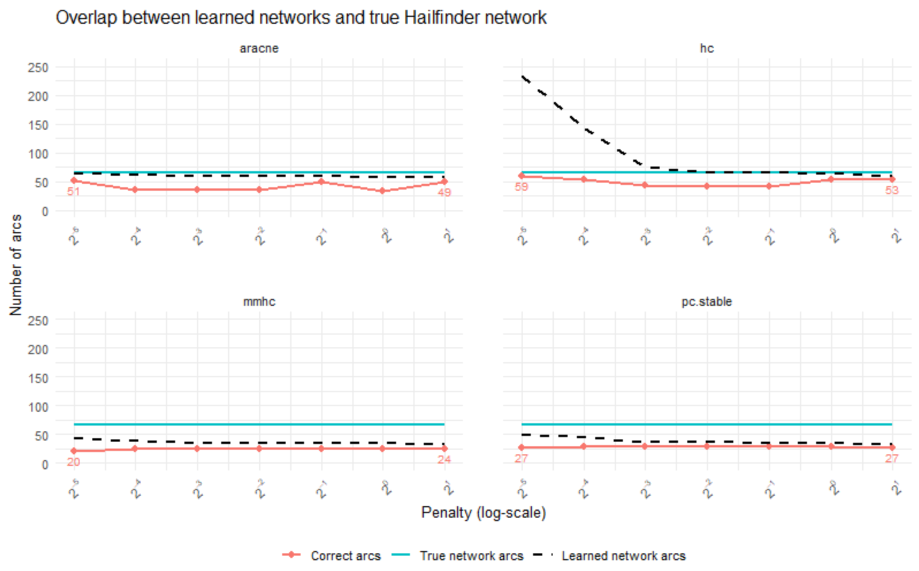


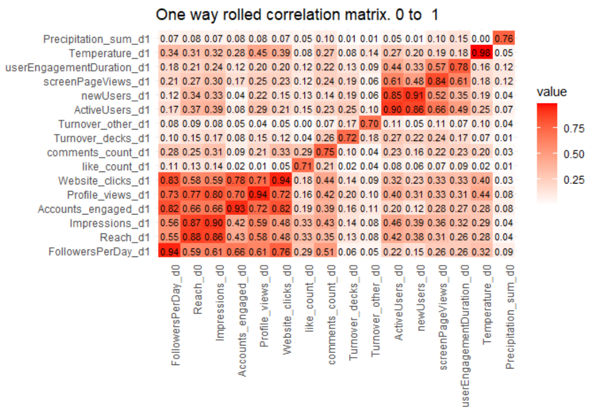
Figure 2: The results of the structure learning process are shown for each algorithm. The correct edges are displayed in red and ideally approximate the 66 true edges that are shown as a reference line in blue. The total amount of edges in the learned networks are shown too to contextualize the cost in terms of false positives of finding the corresponding number true positives.

The cross-correlation analysis was performed on the original stationarized data. Initially, the cross-correlation matrices for several delays. In particular, these matrices were inspected to investigate the behavior of the cross-correlations between smaller and larger delay windows. The cases that were inspected were cross-correlations between single-day windows, weekly windows, and biweekly windows. For the matrices shown in figure 3, the normalized cross-correlations (NCC) between a single non-aggregated row and a rolling window of one, seven and fourteen days are shown. The most visible pattern between these matrices is the fading of the diagonal line as the size of the window increases. Obviously, this implies that autocorrelation is stronger the more recent the observations are. However, it should be noted that some NCC values do become stronger over time. For instance, the block of strong correlations between the Instagram variables in the bottom left corner of each of the matrices is wider for matrices 3b and 3c, compared to matrix 3a. This behavior hints that there are more complex interactions at play in addition to the simple single-day short-term delays.

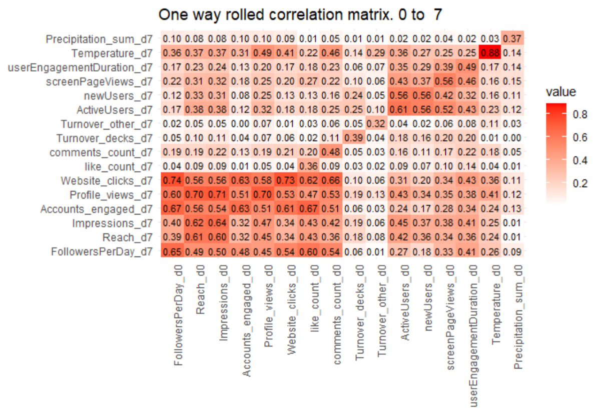
Similarly, the cross-correlations between aggregated windows are displayed in figure 4. These two-way-rolled cross-correlations are computed based on aggregated windows of 7 consecutive days and taking another delay of either 7 or 28 days that precedes these initial 7 days. The NCC values hence represent the cross-correlation between some week and the one or two weeks directly prior, across the entirety of the data set. Interestingly enough, all matrices display some negative correlations, whereas these were completely absent in the previous correlations from figure 3. The most apparent cause for these negative NCC values comes from the precipitation, which makes sense because a greater amount of precipitation naturally leads to fewer people skating and evidently correlates to fewer people engaging with skating content in general. Additionally, the NCC values seem to be smoother between these aggregated windows, compared to the one-way aggregated case.

From figures 3 and 4 it is clear that different behaviors can be observed within the data. Namely, more extreme correlations that are closer to 0 or 1 for short delays and smoother correlation values for between wider delays. To

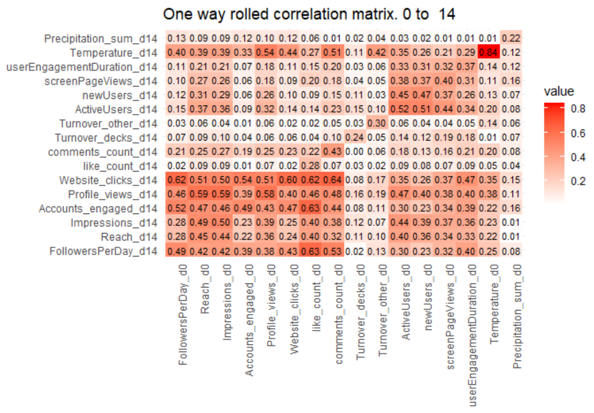
Cross-correlation matrices



(a)



(b)



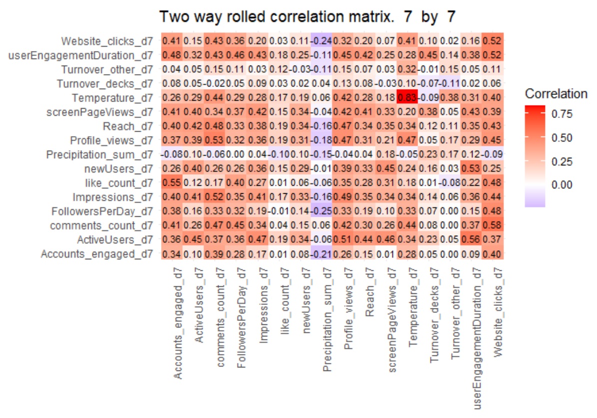
(c)

Figure 3: Correlation matrices displaying the cross-correlations between the single-observation, non-delayed (d0) values and delays of one-day, seven-day or fourteen-day aggregated windows in images (a), (b) and (c) respectively.

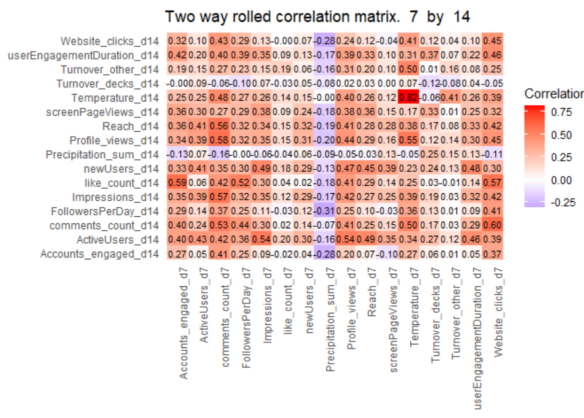
get a complete overview of the NCC values with respect to the different delay configurations, all NCC values were computed for all configurations of delays with a delay shorter than or equal to 28 days. Each of these configurations results again in a matrix similar to the ones discussed before and a mean absolute cross-correlation was computed to represent a total correlation score for a given configuration of delays. The results are presented in figure 5. The highlighted curve in this figure represents the curve that attains the greatest value among these configurations at some window size of interest d_j. The window length d_i of the preceding window results in a maximal MACC for a window length of 7, which coincides with delay window length of the window of interest d_j. This means that the configuration of delays that attains the greatest MACC is given by cross-correlating a one-week aggregate with its preceding week aggregate. This finding motivates the use of delay windows of consecutive weeks. Still, the use of shorter delays might be worthwhile for their stronger potential associations, and we will therefore discuss both of these configurations and use this finding to support the choice of the particular set of delays used. Moving forward, two different delayed data sets will be used, using either delay set $T_d = \{1, 2, 7, 14\}$ or $T_w = \{7, 14, 28\}$ where the subscripts d and w refer to 'day' and 'week'. The corresponding data sets are consequently denoted as $\mathbf{X}_{T_d}$ and $\mathbf{X}_{T_w}$ respectively, and these notations will also be used to reference the configuration of the model setup along with the type of blacklist used.

Using data set $\mathbf{X}_{T_w}$ with the co-occurring blacklist and without model averaging, some initial results were produced. From the resulting graphs shown in figure 6, we can already see the impact of this experimental setup on the results. Judging from the figure, it can be seen that the structures vary greatly, although some edges are shared.

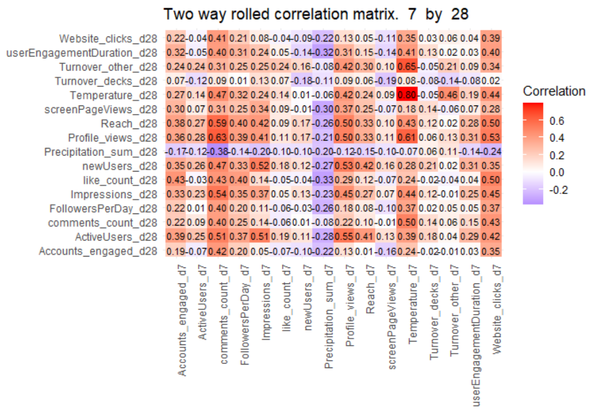
Cross-correlation matrices



(a)



(b)



(c)

Figure 4: Correlation matrices displaying the cross-correlations between the seven-day aggregated window (d7) values and delays of seven-day, fourteen-day and twenty eight-day aggregated windows in images (a), (b) and (c) respectively.

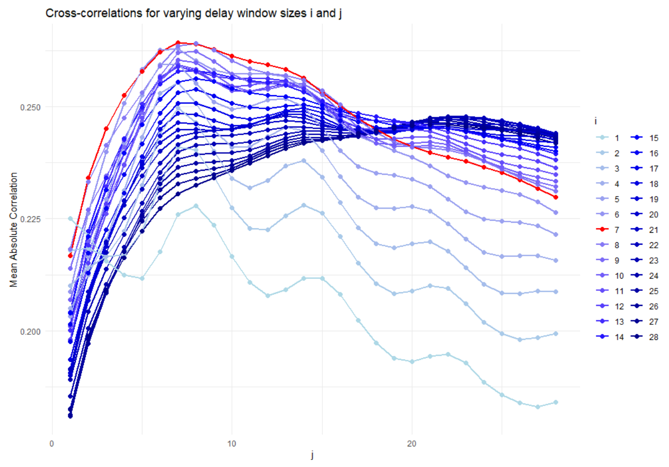
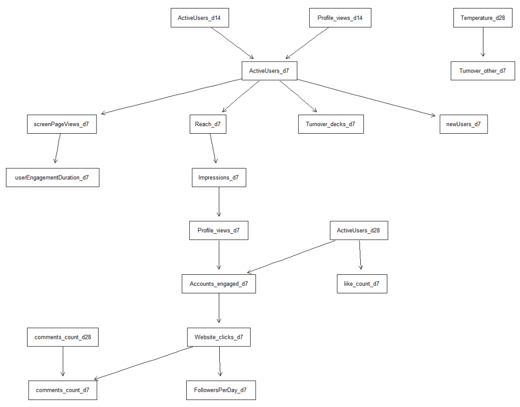
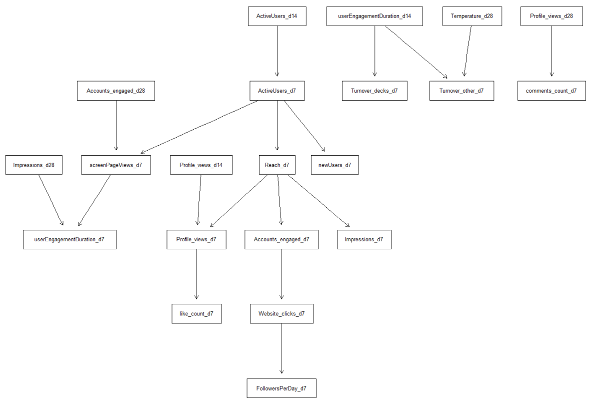


Figure 5: Mean absolute cross correlations (MACC) of all configurations of delays. The MACC values are on the vertical axis and the window size of the period of interest is on the horizontal axis. The different window sizes for the window that precedes the window of interest are represented with different curves. The curve that attained the maximum MACC value among these windows is colored red for clarity. This curve belongs to a preceding window containing seven days and attains its maximum value at a window of interest that also spans seven days (i.e., the value on the horizontal axis is seven as well)

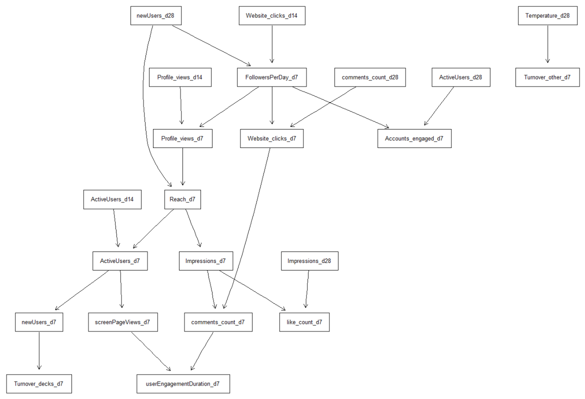
Graphs produced with HC



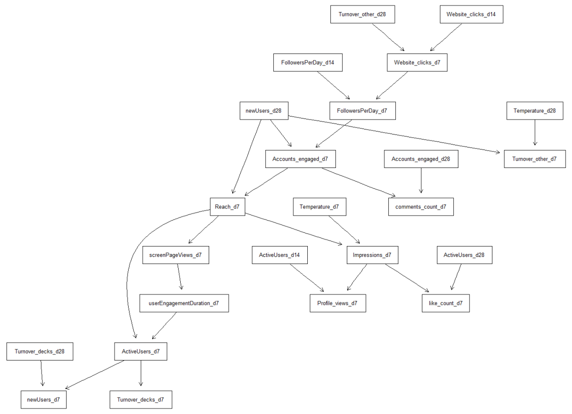
(a) The network learned from $\mathbf{X}_{T_w}^{\text{PAM}}$



(b) The network learned from $\mathbf{X}_{T_w}^{\text{interval}}$



(c) The network learned from $\mathbf{X}_{T_w}^{\text{quantile}}$



(d) The network learned from $\mathbf{X}_{T_w}^{\text{Hartemink}}$

Figure 6: Four Bayesian networks learned from the same data set $\mathbf{X}_{T_w}$ with different discretizations. No bootstrapping and model averaging was applied and the default penalty value of p_0 was used during structure learning.

Comparing the four structures where figure 6a is used as the reference network, table 2 shows how similar the edge structure is between these networks. True positives (TP) are edges shared between the network, false positives (FP) are edges found in the candidate network that is tested against the reference network, and false negatives (FN) are edges that are absent from the candidate network but are found in the reference network. It appears from table 2 that the PAM discretization contains the fewest edges and shares at most half of its edges with any of the other networks. Similarly, the results for the networks that resulted from using the strictly causal blacklists are shown in table 3. These results show a strong difference between the networks. However, similar results were found when the interval discretization network was compared to the other networks. Only the quantile and Hartemink discretization networks appeared to be quite similar when comparing these with the strictly causal blacklist. These results emphasize the impact of the type of discretization.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
PAM vs. interval	9	9	8
PAM vs. quantile	8	15	9
PAM vs. Hartemink	8	17	9

Table 2: Comparison of Bayesian networks trained on different discretizations of the data set $\mathbf{X}_{T_w}$ using the HC algorithm with the co-occurrence allowing blacklist

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
PAM vs. interval	3	13	14
PAM vs. quantile	3	23	14
PAM vs. Hartemink	2	22	15

Table 3: Comparison of Bayesian networks trained on different discretizations of the data set $\mathbf{X}_{T_w}$ using the HC algorithm with the strictly causal blacklist

3.2 Day-level aggregates, $B_{\mathbf{co}}$

This study aims to cover a variety of experimental setups to model data that might be useful in business contexts. With this goal in mind, the upcoming results will be structured into four sections, based on the type of delay aggregates that were chosen in combination with a particular blacklist. Namely, the daily and weekly data aggregates $\mathbf{X}_{T_d}$ and $\mathbf{X}_{T_w}$ will be used in combination with blacklists B_{co} and B_{causal}. Subsequently, the data were discretized according to a PAM, interval, quantile or Hartemink approach. These preprocessing variations were used to learn Bayesian networks using the HC, pc.stable, and hybrid MMHC and ARACNe algorithms. By using these algorithms with a range of structure learning penalty hyperparameters, the performance of each of these models was monitored for each of the discretizations methods. By considering predictive, probabilistic metrics as well as model complexity measuring metrics, a balanced picture of model performance is created. Hence, for each of our four preprocessing configurations, a (set of) candidate network(s) is proposed as the most suitable network(s) to model the data from among the four discretizations.

Starting with data set $\mathbf{X}_{T_d}$, which contains both single-day delays as well as weekly delays, in combination with blacklist B_{co}, the largest combination of circumstances is introduced. Namely, both daily and weekly aggregates are used and these variables are allowed to interact in a co-occurring fashion as described in section 2.2.

Inspecting the plots in figure 7, some interesting patterns can be deduced. Recall that lower Brier scores are favorable. Perhaps the most interesting aspect is that the test results show better performance than the training results in some of the plots. The main contributor to this phenomenon is the type of train-test split that was used. Because a time-respecting split was made, the two splits do not necessarily share a highly similar distribution compared to random sampling. This effect is especially poignant in figure 7b, where the models perform significantly better on the testing data set compared to the training data set. Moreover, the Brier scores are exceptionally low compared to all other discretizations. Reflecting on the data sets resulting from the proposed train-test split, it shows that interval discretization is not suitable for this type of analysis. With the nature of the data at hand, interval discretization is prone to having bins that contain the vast majority of observations. Consequently, this leads to the training split being dominated by the same labels, leading to very high probabilities that this label will occur and that this label will be predicted correctly subsequently. Even more so, the test data set appeared to be even more one-sided in terms of label diversity, leading to inflated Brier scores from the non-informative discretized data set obtained from $\mathbf{X}_{T_d}^{\text{interval}}$.

However, the other discretization methods show more expected behavior with the testing data set performing similarly for all algorithms, except for the ARACNe algorithm. The cause of this is similar to what happens in the models based on interval discretized data. That is, it appears from the black dashed line that the ARACNe algorithm structurally produces relatively small networks, as can be seen in table 27 in appendix A. Because the networks are smaller, the mean Brier scores are more strongly influenced by a few nodes that show more a one-sided presence of labels. Interestingly, it seems that for the preprocessing configuration at hand, the PAM discretization suffers the least from this for the model produced by the ARACNe algorithm. The child nodes were presented alongside the Brier score because the Brier score is only computed for nodes that have a non-empty dependency structure in the network and are a child node inside of it. This allows for assessing the Brier score in context to the number of variables that were scored, such that the model's predictive performance can be weighed against the complexity of its corresponding network.

Disregarding the plot in figure 7b that uses the interval discretization and the sparse ARACNe networks, it appears that the models trained on the PAM discretized $\mathbf{X}_{T_d}$ data set provide the best performance. Interestingly, table 27 shows that the models produced at a penalty value of p_2 are very similar in terms of number of nodes and edges, hinting at a convergent structure. Upon inspection of the plot from the ARACNe algorithm, it appears that it does not contain any edges that represent a dynamical structure. Rather, it only contains edges that represent co-occurring relations. Comparing the edges present between the three remaining networks in table 4, it can be seen that the networks indeed show a largely similar structure. Judging from table 27, it appears that the Brier, accuracy and AUC scores are very similar. Meanwhile, the model complexity in terms of nodes and edges does vary between the pc.stable and the HC and MMHC algorithms. This is also reflected in table 4, where the networks learned from HC and MMHC are more similar to each other than similar to the network learned from the pc.stable algorithm.

However, regardless of having convergent structures at large penalties, these models are not necessarily optimal in terms of predictive performance. Table 27 shows that all of the predictive performance metrics display an increasing performance as the complexity of the model decreases. Therefore, checking the BIC values for these models helps to visually inspect which model and what penalty level produce models that provide the best balance between predictive performance and model complexity. The BIC results are displayed in figure 8. BIC scores are shown for the networks learned during both the training and the testing phase so that the stability of the network could be gauged. The BIC scores from the testing results are used for assessing and comparing networks in the tables found in the appendices. From this figure, we can see that the BIC scores for the models presented in table 4 are part of a downward trending graph, implying that opting for a network from a smaller penalty with greater model complexity would be beneficial in terms of BIC score. From this, it could be argued that using a PAM discretization alongside the HC algorithm applied with a penalty of 0.25 would achieve the best BIC score, while only marginally compromising on average predictive probabilities. However, MMHC algorithm has slightly simpler models with fewer edges and has a non-increasing BIC score across penalties, implying that the best fit is obtained at the lowest penalty level. Judging from table 27, it can be seen that the MMHC algorithm provides a network at penalty p_{-7} with similar amounts of nodes and edges to the HC algorithm at p_{-2}. The same holds for the pc.stable algorithm at the minimum penalty level. However, the maximum path length of the HC algorithm model is significantly larger, boasting a value of 7, compared to a length of 4 in both pc.stable and MMHC. Upon comparing the edges present in these networks, they appear to be relatively similar. In particular, 13 out of the 15 edges from the MMHC network are present in the pc.stable network, which contains 17 edges. All in all, given that the MMHC model is slightly more parsimonious and has slightly better predictive performance than the pc.stable model, the MMHC model will be set as the preferred model for now. We note that this similarity between the MMHC and pc.stable networks decreased upon inspecting the edges present in the pc.stable generated network at penalty level p_{-5}, where the number of nodes and edges differed only by one from the previous MMHC model. This implies that the MMHC and pc.stable algorithms diverge slightly at the intermediate penalty levels investigated, before converging to a similar structure at the penalty level of p_0 and above. Moreover, comparing the results from

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
pc.stable vs. HC	6	2	3
pc.stable vs. MMHC	6	1	3
pc.stable vs. ARACNe	0	4	9
HC vs. MMHC	7	0	1

Table 4: Comparison between networks learned from different algorithms at a penalty level of p_2.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
HC p_0 vs. ARACNe p_{-7}	6	2	3
HC p_0 vs. MMHC p_0	6	1	3
HC p_0 vs. pc.stable p_0	0	4	9

Table 5: Comparison between the reference HC network and the other algorithms with PAM discretization. Networks from the remaining algorithms were selected based on similarity in edge count.

table 27 against tables 31 and 33, the results from the PAM discretization offer slightly, but consistently better predictive performance on the averaged Brier and AUC scores.

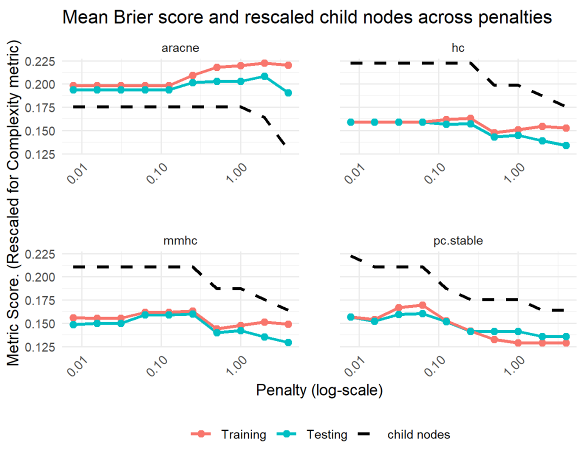
In contrast to using predictive performance, using the results from the preliminary analysis on the Hailfinder data set, a different network would be deemed appropriate. One could argue that the network learned by the HC algorithm at the standard BIC corresponding penalty level p_0 can be used as the reference network. From this perspective, the networks learned from the other algorithms can be compared to see if the methods show comparable results and improve support for having a robust network structure. The comparisons between the reference network and the networks from the other algorithms are shown in table 5.

Ultimately, the MMHC network is displayed in figure 9, where only connected nodes are displayed to prevent visual clutter. This network has some noteworthy structures. Namely, it appears that a dependency structure between data sources, such as Google Analytics and Instagram barely present in this network. In particular, it contains one edge between the amount of active users and the co-occurring amount of comments. The direction of this edge is favored slightly in this direction. That is, with a strength of 0.992, this edge appeared in this direction around 54% of the time across the models obtained from the bootstrapping process. This ambiguity is plausible because people visiting the web shop can be referred to the Instagram page, where they can make comments on posts. Conversely, it might also make sense for this edge to be reversed because people might get referred to the web shop from Instagram rather than the converse.

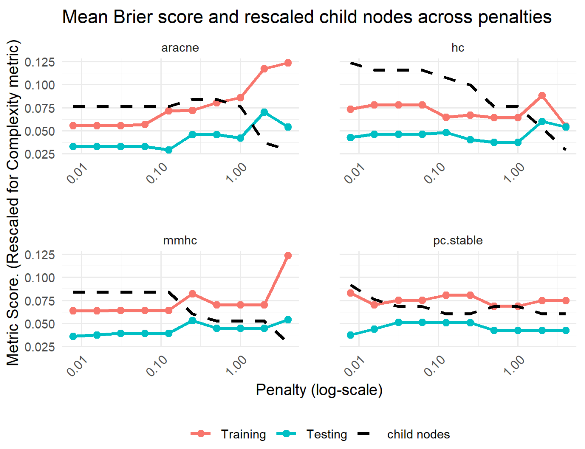
Another property of this graph is that it consists of multiple subgraphs. These subgraphs display autoregressive, dynamic relationships within a variable along with co-occurring relationships between different variables. Some edges represent a strictly dynamical interaction, such as for Accounts_engaged_d2 and comments_count_d2; or a co-occurring relationship, such as for ActiveUsers_d1. In case of the Accounts_engaged_d2 and comments_count_d2 variables, they are part of a v-structure along with another co-parent that represents a different variable altogether. These structures imply that the child node is best explained using both a past observation of itself alongside a different variable. However, such v-structures are a minority class inside all the dependency structures inside of the network. The other dependency structures represent only single parent structures that are either co-occurring or dynamic in nature.

In the same vein, the reference HC network is shown as well in figure 10. This network shows several similarities with figure 9, as expected from the convergent structures found and the comparison table, but has fewer nodes of higher degree and fewer arcs overall. With similar predictive performance according to table 27, and acknowledging the HC network as the reference network, this suggests the MMHC network could be comparatively overfitting. Moreover, 9 out of the 15 arcs of the MMHC network overlap with the HC network. Although this shows a substantial

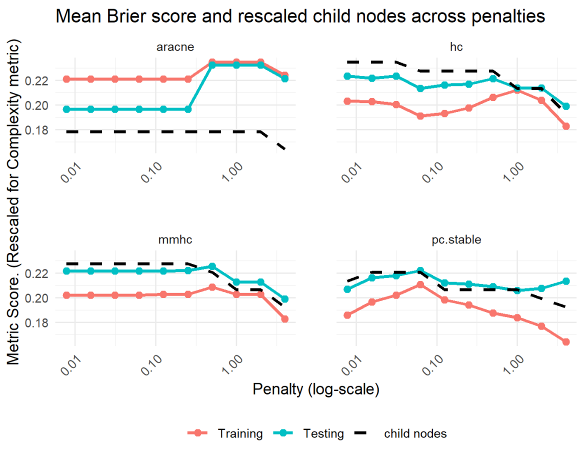
Brier score performance plots, $\mathbf{X}_{T_d}$-$B_{\mathbf{co}}$



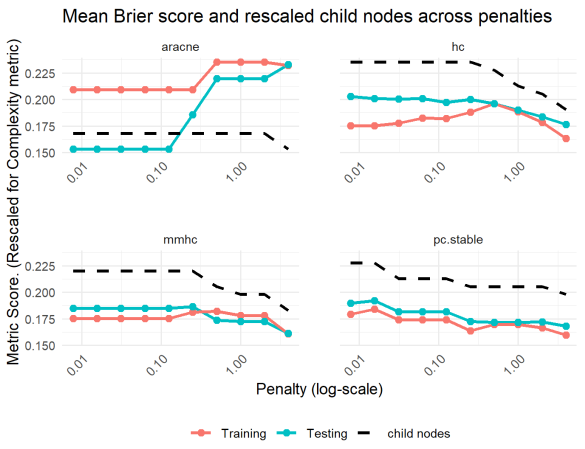
(a) PAM discretization



(b) Interval discretization



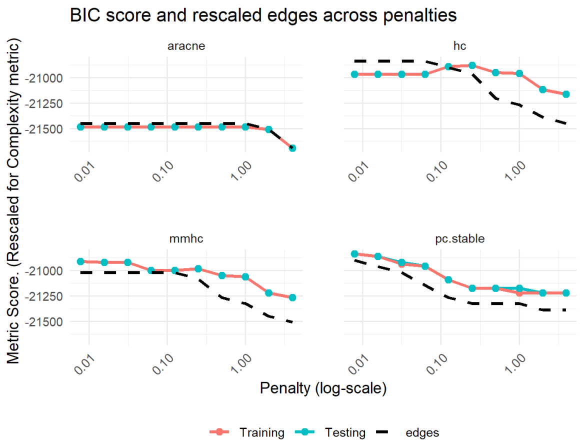
(c) Quantile discretization



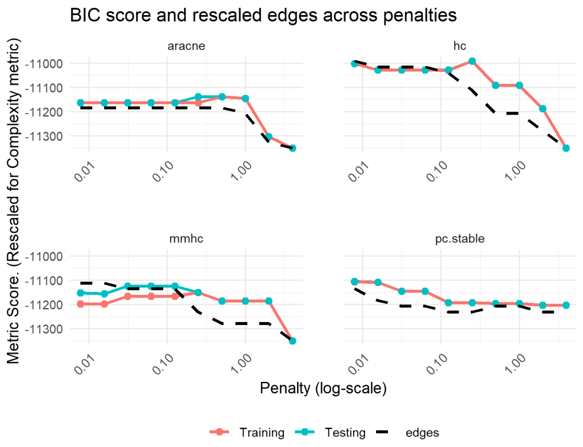
(d) Hartemink discretization

Figure 7: Mean Brier score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of child nodes present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

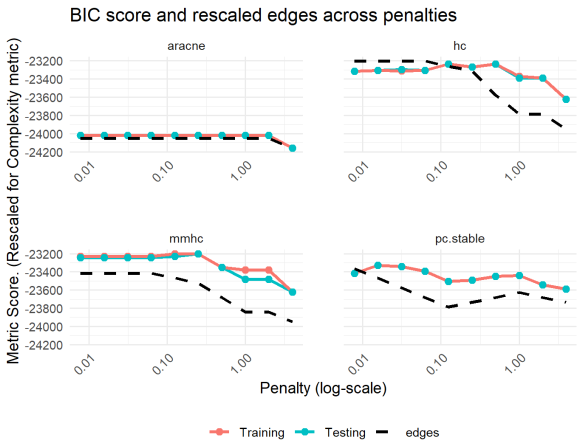
BIC score plots, $\mathbf{X}_{T_d}$-$B_{\mathbf{co}}$



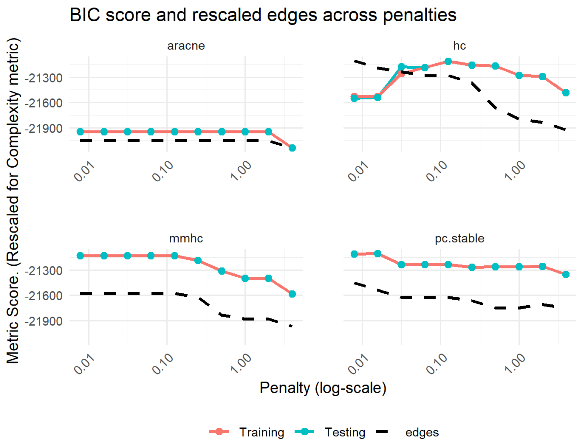
(a) PAM discretization



(b) Interval discretization



(c) Quantile discretization



(d) Hartemink discretization

Figure 8: BIC score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of edges present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

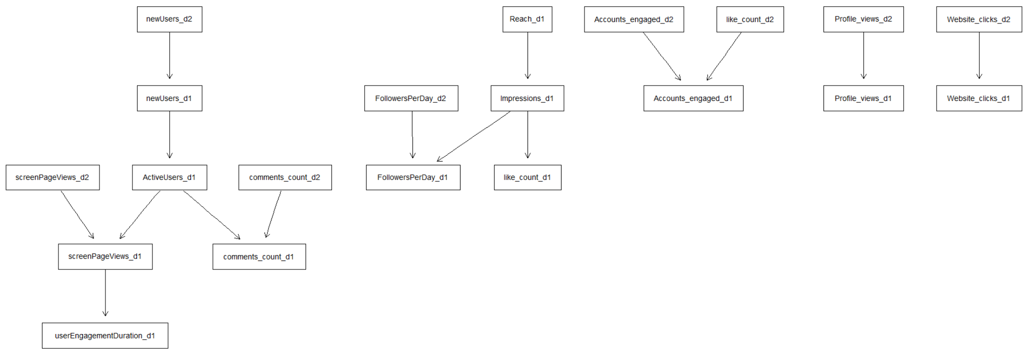


Figure 9: The network produced by the MMHC algorithm, learned from the data $\mathbf{X}_{T_d}$ that was subjected to PAM discretization and using blacklist B_{co} and penalty p_0. Nodes that were assigned no edges in any direction are omitted from the graph.

amount of overlap, a significant amount of edges differ between the networks. This highlights that chasing predictive performance can lead to significant differences in the network structures compared to the reference network.

Lastly, the strengths of the edges were inspected to see how robust the edges were under bootstrapping. The strengths of the edges provide a measure of how likely all of the possible edges would be included in the final model. The strength plot is presented in figure 11. In this plot, there is a clear cluster near both extremes of the strength range. The vast majority of edges are on the left, clustered within $[0.0; 0.2]$. This makes sense because among all of the possible edges, of which there are many, only the most powerful few should end up in the final model. Consequently, the edges that are not a part of this weak cluster are spread out across the remainder of the range. Only 8 of these remaining edges had a strength that did not meet the 0.75 threshold. Moreover, 13 out of the 15 edges in the final model had a strength greater than 0.93, which implies that the model consistently found these edges in nearly all of the models resulting from the bootstrapped data sets. This gives great confidence that these edges are relevant. As for the reference HC network, the results were similar, but showed fewer cases of moderate strength as shown in figure 12.

Lastly, for completeness' sake, the networks learned from the PAM and Hartemink discretization were compared to see if the different discretizations elicit different relationships. The results are shown in table 6. The models for each algorithm used for these comparisons were selected based on their overall performance within the discretization method, excluding the ARACNe-HC results for its sparse graphs. These results were obtained from tables 27 and 33. The table highlights some interesting findings. Comparisons within discretization methods seem to have the most overlap in most cases. Only in comparisons including networks learned from the HC algorithms are large differences observed. This because these networks typically have larger edge counts. Interestingly, some cross-comparisons, such as comparison MMHC_{PAM} p_{-7} vs. $\text{MMHC}_{\text{Hart}}$ p_{-3} show relatively similar networks as well. Based on the false positives and negatives, the overall precision and recall are 0.6505818 and 0.6806714 respectively, as shown in table 7. The other results in the table show that the Hartemink discretization provides slightly more consistent networks between the different algorithms compared to the PAM discretization. The between-discretization comparison shows the weakest results. However, the precision and recall show that majority of the edges are shared between such networks on average.

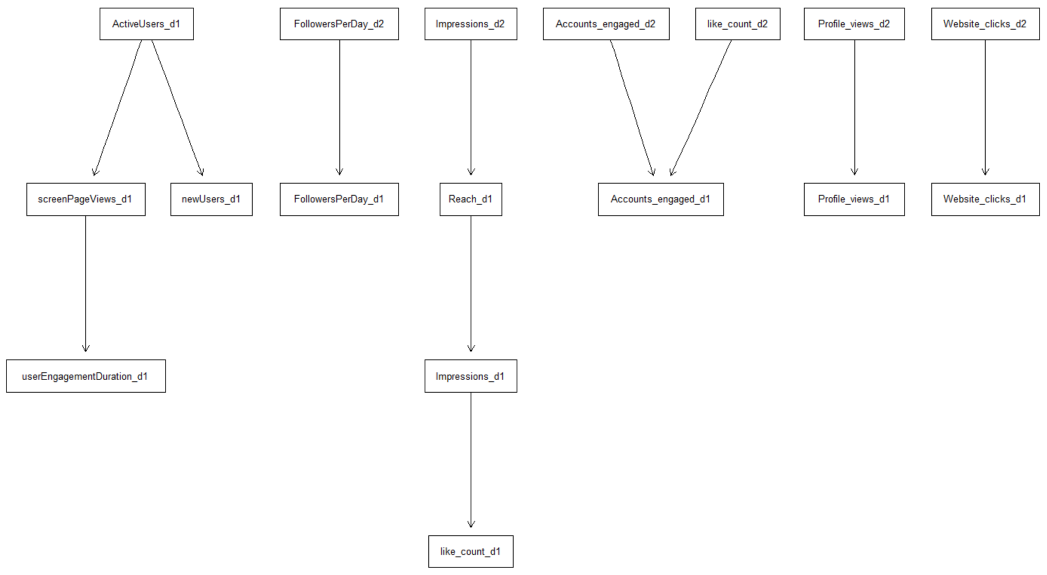


Figure 10: The network produced by the reference HC algorithm, learned from the data $\mathbf{X}_{T_d}$ that was subjected to PAM discretization and using blacklist B_{co}. Nodes that were assigned no edges in any direction are omitted from the graph.

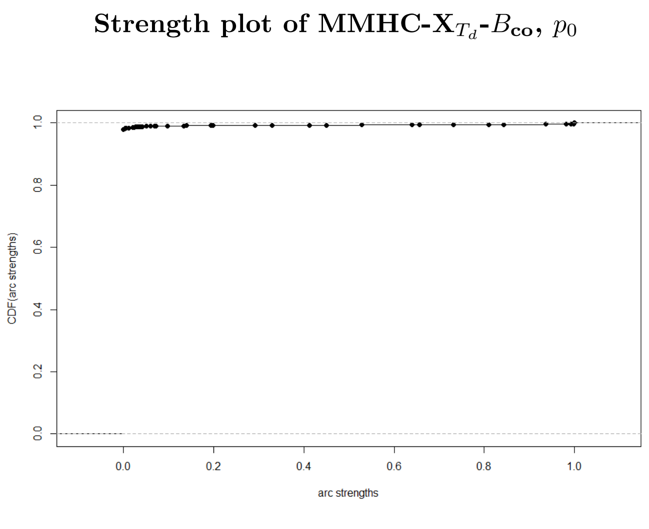


Figure 11: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge across the models obtained from all of the bootstrapped data samples.

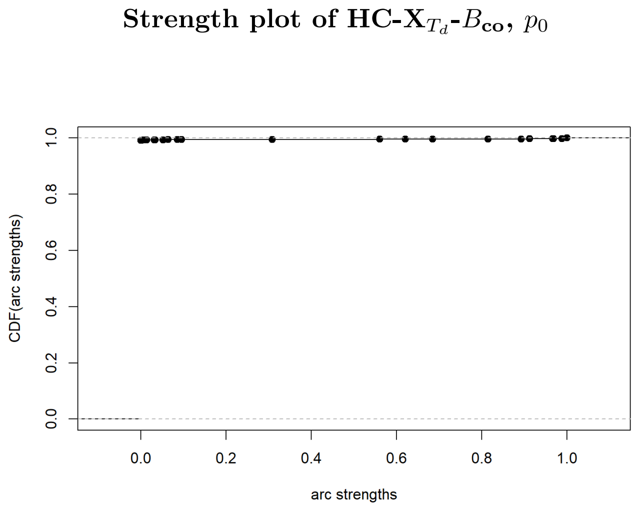


Figure 12: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge across the models obtained from all of the bootstrapped data samples.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
pc.stable$_{\text{PAM}}$ p_{-7} vs. HC$_{\text{PAM}}$ p_{-2}	10	6	7
pc.stable$_{\text{PAM}}$ p_{-7} vs. MMHC$_{\text{PAM}}$ p_{-7}	13	2	4
pc.stable$_{\text{PAM}}$ p_{-7} vs. pc.stable$_{\text{Hart}}$ p_{-6}	10	7	7
pc.stable$_{\text{PAM}}$ p_{-7} vs. HC$_{\text{Hart}}$ p_{-3}	9	14	8
pc.stable$_{\text{PAM}}$ p_{-7} vs. MMHC$_{\text{Hart}}$ p_{-3}	11	5	6
HC$_{\text{PAM}}$ p_{-2} vs. MMHC$_{\text{PAM}}$ p_{-7}	11	4	5
HC$_{\text{PAM}}$ p_{-2} vs. pc.stable$_{\text{Hart}}$ p_{-6}	9	8	7
HC$_{\text{PAM}}$ p_{-2} vs. HC$_{\text{Hart}}$ p_{-3}	11	12	5
HC$_{\text{PAM}}$ p_{-2} vs. MMHC$_{\text{Hart}}$ p_{-3}	10	6	6
MMHC$_{\text{PAM}}$ p_{-7} vs. pc.stable$_{\text{Hart}}$ p_{-6}	11	6	4
MMHC$_{\text{PAM}}$ p_{-7} vs. HC$_{\text{Hart}}$ p_{-3}	10	13	5
MMHC$_{\text{PAM}}$ p_{-7} vs. MMHC$_{\text{Hart}}$ p_{-3}	12	4	3
pc.stable$_{\text{Hart}}$ p_{-6} vs. HC$_{\text{Hart}}$ p_{-3}	15	8	2
pc.stable$_{\text{Hart}}$ p_{-6} vs. MMHC$_{\text{Hart}}$ p_{-3}	15	1	2
HC$_{\text{Hart}}$ p_{-3} vs. MMHC$_{\text{Hart}}$ p_{-3}	13	3	10

Table 6: Comparison between networks learned from different algorithms at various penalty levels, as described in the model names.

Comparison	Precision	Recall
PAM	0.7416667	0.6801471
PAM - Hartemink	0.5701726	0.6488562
Hartemink	0.8007246	0.7766411
Overall	0.6505818	0.6806714

Table 7: Precision and recall values for when aggregating different comparisons within or between discretizations

3.3 Day-level aggregates, B_{causal}

To complement the results previously displayed, the networks that were generated using the strictly causal blacklist, B_{causal}, will be discussed. Notably, the ARACNe-HC algorithm produces empty networks for all discretizations within this configuration, which means that the penalization used in combination with B_{causal} prevented the remaining candidate edges from being labeled as significant. Moreover, upon inspection of the graphs resulting from the hybridized ARACNe-HC algorithm with blacklist B_{co}, it showed that all the networks contained no edges that represented dynamical relationships to begin with. Therefore, the ARACNe-HC algorithm will not be considered when discussing or presenting the results that were produced in this X_{T_d}-B_{causal} experimental setup.

For the other algorithms, we can see similar results with this strictly causal setup compared to the co-occurrence setup. Namely, the interval discretization overperforms again because the resulting bins are imbalanced to the extent that an overwhelming amount of observations are labeled similarly within the testing data set. Additionally, the Hartemink discretization consistently outperforms the quantile discretization in the respective testing data sets when comparing the same algorithms, regardless of penalty level. Therefore, the best-performing models probably originate from either the PAM or Hartemink discretizations. First of all, it can be observed from both figure 16a and table 28 that while the child nodes appear to be relatively constant for all algorithms for most of the penalty values, the corresponding amount of edges decreases for the pc.stable and HC algorithms at a rate similar to the models created from the blacklist B_{co}. This means that the network retains the same child nodes at different penalization levels and increasing penalization results in pruning edges that share a child node. The MMHC algorithm appears to have a network structure that is very robust to increasing penalization. Moreover, as soon as the networks of this algorithm start to become less complex starting from a penalty greater than 2, the BIC score also starts to decrease, showing that the overall fit becomes less appropriate when the network becomes simpler and the initial network could not be improved by pruning the edges.

Secondly, the performance of the HC algorithm shows some interesting patterns as well. The HC algorithm's mean Brier score and its number of child nodes from figure 16a show that the number of child nodes is stable for a long time, while table 28 shows that the amount of edges does decrease. This means that the average degree of nodes also decreases, as shown in the table. Similarly, this can be interpreted as the same number of variables predicted with a decreasing number of predictors. Inspecting the Brier score, the figure shows that there is no clear monotone pattern in the Brier score, suggesting a network from an intermediate penalty level results in the best probabilistic performance. However, figure 17a shows that the BIC score is optimal at the penalty level where the number of child nodes has not decreased yet, and starts to decrease when the number of child nodes decreases. In contrast, looking for a penalty level to optimize Brier score performance results in a model that is overfitting relative to a simpler model in terms of BIC. Hence, this indicates that lower Brier scores could also originate from overfitting and should be treated with caution.

Similarly, the results for the pc.stable algorithm display a similar pattern. The patterns are less obvious from the plots, but can be read from table 28. In particular, the Brier scores tend to decrease as the penalty level increases and the model complexity reduces, but the BIC value peaks at a penalty level that does not advocate for the lowest Brier score attained at the higher penalty values.

The networks learned from the HC and pc.stable algorithm can be used to contextualize the results from the robust MMHC results. These networks learned from the MMHC algorithm, except for the last two penalties of values p_1 and p_2, have 15 nodes and 8 edges. Consulting table 28, some comparable models from other algorithms are the pc.stable algorithm at penalty level p_{-2} and the HC algorithm at penalty level p_1. Comparing edges used in these networks in table 8, it can be seen that the best MMHC model has a lot of overlap with the pc.stable and HC algorithms for some particular penalties. However, these pc.stable and HC models at these penalty levels are not the optimal models within their own algorithm results. Therefore, we will conclude that the MMHC network, although robust, might give a too narrow representation of the data. In contrast, the pc.stable and HC algorithms provide

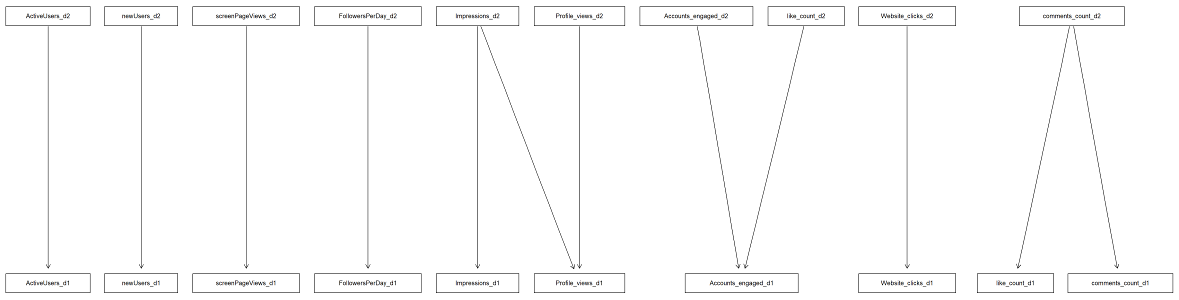


Figure 13: The network produced by the pc.stable algorithm, learned from the data $\mathbf{X}_{T_d}$ that was subjected to PAM discretization and using blacklist B_{causal}. Nodes that were assigned no edges in any direction are omitted from the graph.

the best BIC scores at penalty levels p_{-3} and p_{-1} respectively, with 12 and 13 edges. Moreover, these models boast similar performance on the Brier score, accuracy and the precision-recall AUC. In addition, the networks' average node degree and entropy are also very similar. The Kullback-Leibler divergence (Scutari, 2024) is used to quantify how well the distributions corresponding to the fitted networks approximate each other. The results are presented in table 9. The pc.stable network provides a lower KL-divergence with respect to the HC network than its converse. This implies that the pc.stable network is probably the better model because it retains more information about the data.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
pc.stable p_{-2} vs. MMHC p_0	7	1	2
HC p_1 vs. MMHC p_0	7	1	2
pc.stable p_{-2} vs. HC p_1	7	2	2

Table 8: Comparison between networks learned from different algorithms applied to the PAM discretization at different penalty levels.

Comparison	KL-value
KL(pc.stable p_{-3}, HC p_{-1})	0.72688
KL(HC p_{-1}, pc.stable p_{-3})	0.81919

Table 9: Comparison of the Kullback-Leibler divergence between two networks.

From this reasoning, the more parsimonious model for the X_{T_d}-B_{causal} setup is the network resulting from the pc.stable algorithm with penalty level p_{-2}. Note that the structure of the network shown in figure 13 is markedly simpler than its counterparts from figures 9 and 10. The most notable difference is the path lengths in the strictly causal network. Because no co-occurring associations are allowed, nor associations between nodes corresponding to windows that do not represent the current or most recent window (d1 in this case), the only remaining edges that are possible are edges that point directly into such a d1-variable, starting from a node with a different time lag.

Let us similarly display the reference HC network. This network did not achieve the best BIC value across penalties, but should still be taken into consideration because it can tell us something about the differences between the reference network and the network at the highest BIC value. The network is shown in figure 15. As previously hinted at by the comparison table, the network structures appear to have great overlap. That is, 10 out of the

12 edges of the pc.stable network appear in the 11 edges found in the reference HC network. In contrast to the previous case where blacklist B_{co} was used, B_{causal} appears to result in more robust network structures across structure learning algorithms. Ultimately, the most appropriate network is either the reference HC network or the pc.stable p_{-3} network, depending on whether the better BIC score should be favored over the reference network or not.

Additionally, upon cross-checking the results with the Hartemink discretization results, some interesting results were observed. In particular, the reference HC network had a similar size with 12 edges, compared to its PAM counterpart having 11. Moreover, between these two differently discretized data sets, the resulting reference HC networks shared 8 edges. Although this still amounts to a meaningful discrepancy between the networks, an interesting find was that when relaxing the penalty factor for the Hartemink based networks, they overlap would increase to 10 edges. However, this would come at the cost of a lower relative overlap approaching 50%. This implies that the different discretization methods can capture similar dependencies, but that different associations are captured at a significant rate.

Revisiting the network from figure 13, the restrictions imposed by the B_{causal} blacklist give rise to some notable observations. That is, the network consists mostly of small subgraphs of two nodes that represent a dynamic autoregressive relationship. The only variables that present other associations, are Profile_views_d1, Accounts_engaged_d1 and like_count_d1. All of these nodes have parents that do not represent their respective past window of observations. Another peculiar association shown in this figure is that the like count apparently has a stronger association with the comments_count_d2 variable than its autoregressive like_count_d2 variable.

The most important observation in this network is perhaps the delays that are used in the parent variables. We reiterate that the variables present in X_{T_d} correspond to the initial 16 variables, subjected to delay set $T_d = \{1, 2, 7, 14\}$. This means that associations originating from parents with a window corresponding to the aggregation of the first or second prior week were also allowed in the strictly causal setting. However, no such associations were found in the structure learning process. This implies that associations between the current row observations and its previous observation for the child nodes in the network are stronger than associations between this current row and either of the week aggregates. Moreover, none of these larger windows could provide an additional association that would be significant under the influence of one of the short-term delays.

Lastly, the model presents very few associations between platforms. Identical to figure 9, there is a relation between the Instagram variable like_count_d2 and the Google Analytics variable Accounts_engaged_d1. Therefore, this model also predominantly presents relationships within the given data sources, rather than between data sources.

The strength plot in figure 18 shows similar behavior to its co-occurrence counterpart from figure 11. There are some clusters at both extremes with the vast majority of edges residing in the strength range between 0 and 0.25. Between 0.25 and 0.75, there are four edges with such intermediate-strength edges. The remaining 12 edges have a strength greater than the 0.75 threshold, of which 7 have a strength greater than 0.99. Interestingly, the strength plot from the reference HC network in figure 19 shows more strongly convergent behavior. This can be seen from the fact that the spread of edge strengths tends more towards the extremities.

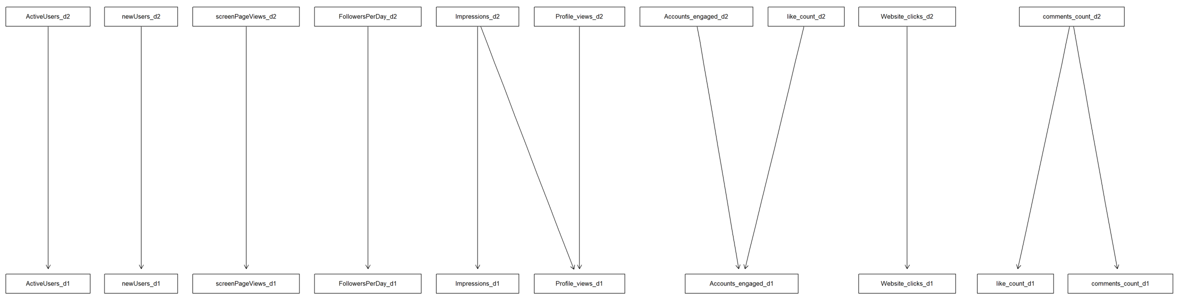


Figure 14: The network produced by the pc.stable algorithm with penalty level p_{-3}, learned from the data $\mathbf{X}_{T_d}$ that was subjected to PAM discretization and using blacklist B_{causal}. Nodes that were assigned no edges in any direction are omitted from the graph.

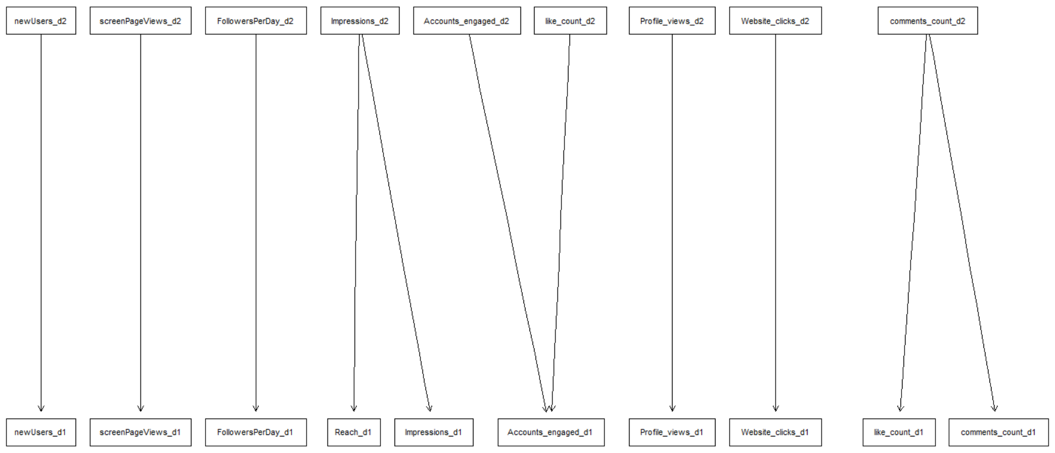
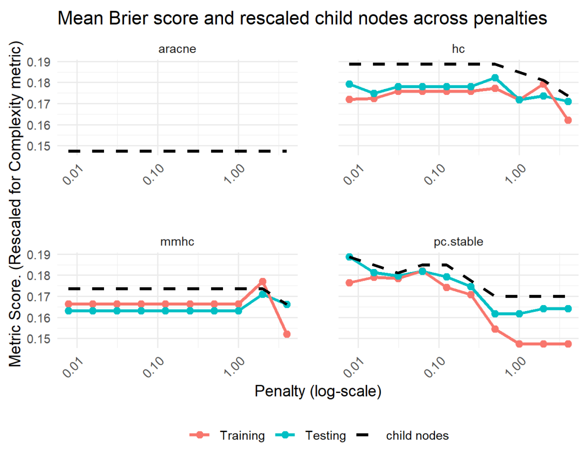
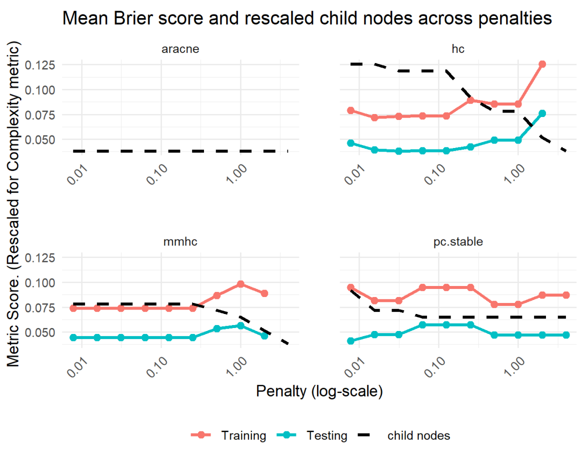


Figure 15: The reference HC network produced with penalty level p_0, learned from the data $\mathbf{X}_{T_d}$ that was subjected to PAM discretization and using blacklist B_{causal}. Nodes that were assigned no edges in any direction are omitted from the graph.

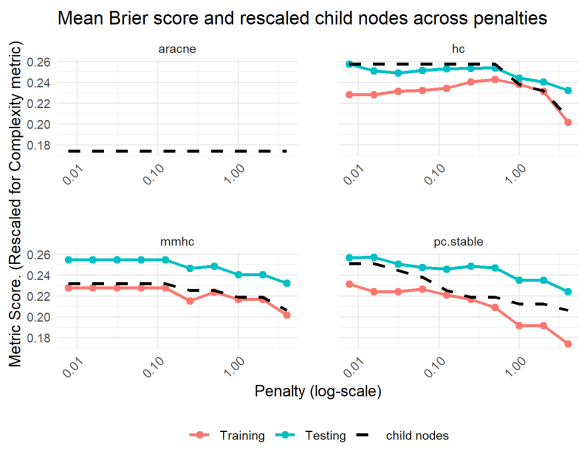
Brier score performance plots, $\mathbf{X}_{T_d}$-$B_{\mathbf{causal}}$



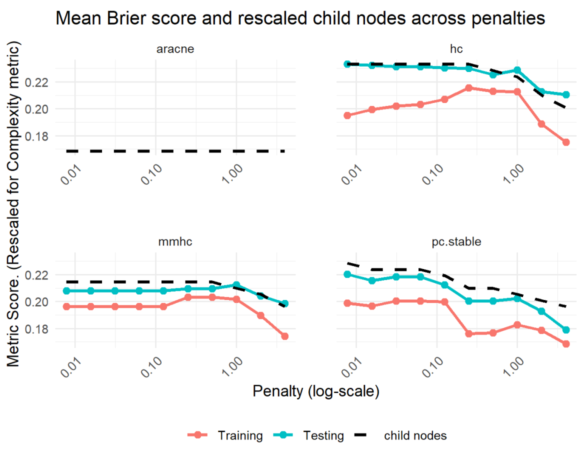
(a) PAM discretization



(b) Interval discretization



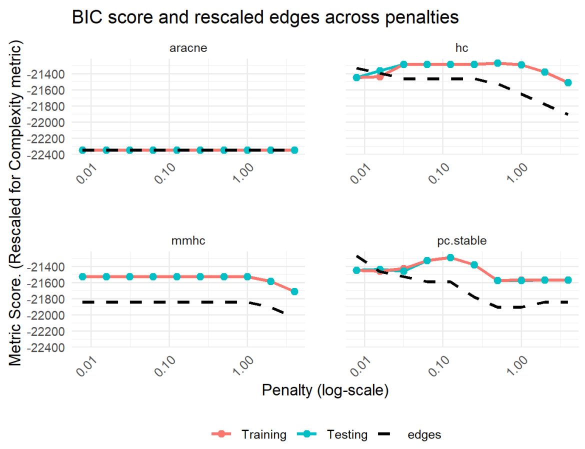
(c) Quantile discretization



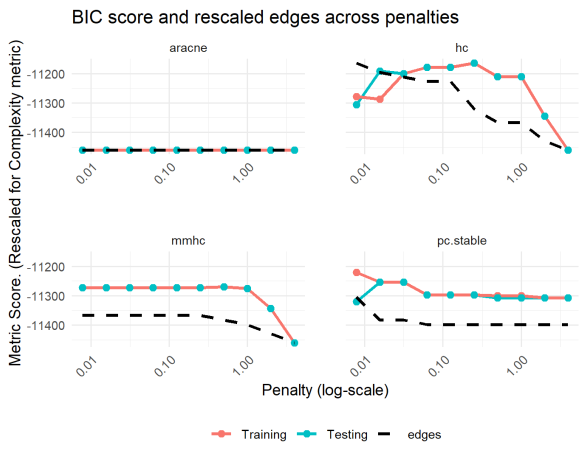
(d) Hartemink discretization

Figure 16: Mean Brier score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of child nodes present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

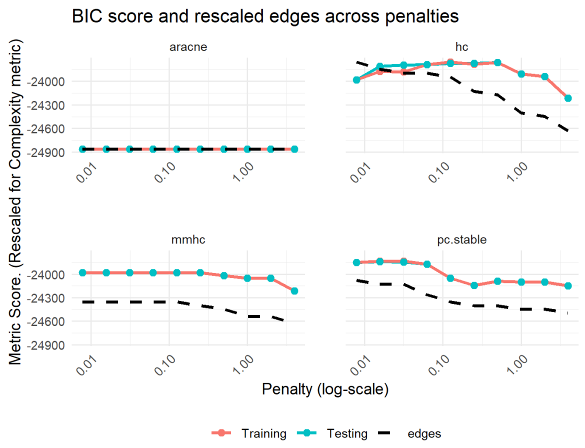
BIC score plots, $\mathbf{X}_{T_d}$-$B_{\mathbf{causal}}$



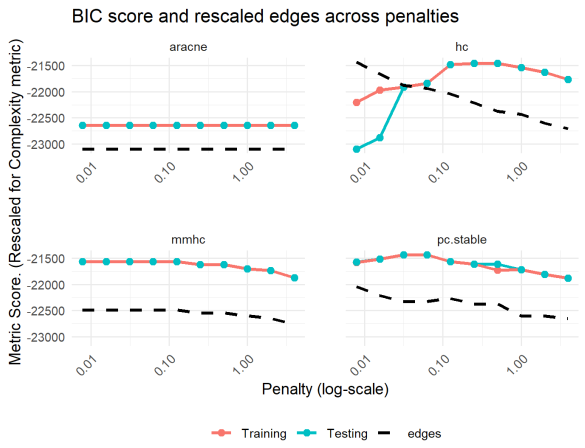
(a) PAM discretization



(b) Interval discretization



(c) Quantile discretization



(d) Hartemink discretization

Figure 17: BIC score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of edges present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

Strength plot of pc.stable-$\mathbf{X}_{T_d}$-$B_{\mathbf{causal}}$, penalty $= p_{-3}$

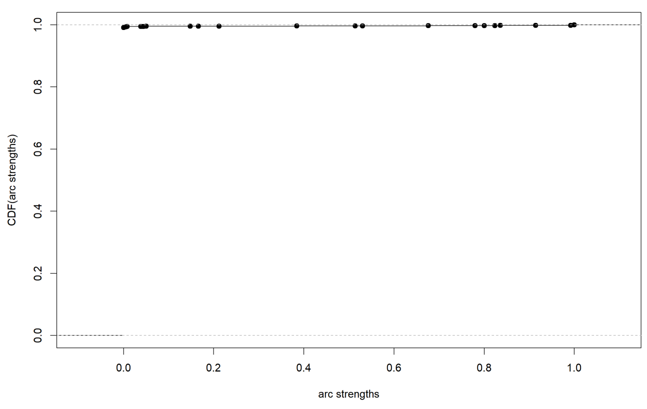


Figure 18: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge presence across the models obtained from all of the bootstrapped data samples.

Strength plot of HC-$\mathbf{X}_{T_d}$-$B_{\mathbf{causal}}$, p_0

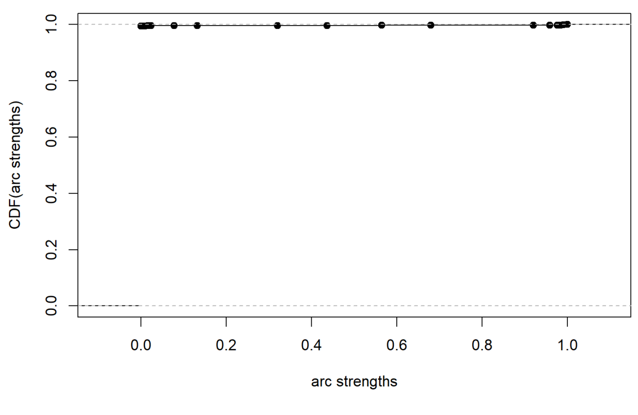


Figure 19: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge presence across the models obtained from all of the bootstrapped data samples.

3.4 Week-level aggregates, $B_{\mathbf{co}}$

The results of the second experimental condition of the aggregation windows will be presented next. In line with the finding from figure 5, time delay set $T_w = \{7, 14, 28\}$ will be used. The probabilistic performance in terms of the Brier score is displayed in figure 20. Similarly to the previous results, the models based on the interval discretization method perform unrealistically well because of the too simplistic bin structure. Additionally, the models based on the quantile discretized data produce structurally worse results than the models based on the Hartemink discretized data. Interestingly, the results for the models based on the PAM discretized data tend to show fewer signs of overperformance based on the fact that the models' performances on the test data seem to structurally be slightly worse than the performance on the training data, as is generally expected. Another interesting initial observation is that the models based on the Hartemink discretized data tend to show more similar performances between the train and test samples compared to the models based on the PAM discretized data. Disregarding plots 21b and 21c, the remaining plots in figure 21 do not show clear optima for the BIC score, and most algorithms show a relatively stable curve. Therefore, the range of penalties was extended to check if the optimum value is within the current penalty range or resides in a higher or lower degree of penalization. Note that due to the formulation of the α threshold, the constraint-based models use an α threshold for the conditional independence test that is at most 0.05. This implies that the minimal penalty value of p_{-7} corresponds to an α value of 0.05, with each subsequent penalty level halving this threshold. Consequently, applying the lower penalties would result in higher thresholds. For that reason, the optimal penalty level for the constraint-based (parts of the) algorithms does not produce strongly statistically significant additional dependency structures. The results for these extended penalty values are displayed in tables 35 and 36 for the PAM and Hartemink discretizations, respectively. Table 35 shows that only the MMHC and ARACNe-HC networks achieve better BIC values at increased α thresholds. However, even with the increased flexibility in the structure learning process, their mean predictive performance metrics just barely compare to the HC and pc.stable networks that are obtained within the initial penalty range. Similarly, table 36 shows that only the MMHC algorithm shows a minor improvement in terms of BIC at the cost of a minor decrease in overall accuracy and the pr-AUC. Therefore, increasing the p-value, which is a debatable approach in and of itself, has some benefits with regard to the overall fit of the resulting networks, but does not improve predictive performance.

Upon inspection of the BIC scores for the HC algorithm, it appears that the highest value resided within the original penalty range for PAM and Hartemink discretizations. Moreover, the Brier score also did not appear to show any notable improvements. This shows that simpler networks do not necessarily produce better results for the few nodes that are included, and the same holds true for more complex networks that resulted from smaller penalties. The results can be seen in tables 19, 25, 35 and 36.

Now, consulting figure 21 and table 19 shows that the best performing algorithms on this discretization are among low to intermediate penalty levels. For the MMHC and ARACNe-HC algorithms, the BIC scores for these two models seem to drop when the penalty becomes strong enough to prune the first edges from the network that was found at the first penalty level, corresponding to an α threshold of 0.05. This implies that the edges are very robust and are only omitted from the network when the penalty becomes sizable. Moreover, pruning the initial network based on increasingly penalized HC-searching detracts from the initial BIC score, showing that the initial network was optimal to begin with within the context of these algorithms. Regarding the pc.stable and HC algorithms, the best results are arguably obtained at penalty levels p_{-6} and p_{-1} respectively. In addition, the reference HC model at penalty level p_0 seems similar in terms of complexity to the ARACNe and MMHC models. Comparing the network structures in table 10, some interesting results appear. Firstly, the HC p_{-1} model shows many true positives with the other models, at the expense of many false negatives. Given that this network contains the largest number of edges compared to the other networks in the table, the large amount of false negatives is explained. This behavior implies that the HC algorithm is able to find networks that largely overlap with networks from other algorithms, but includes additional edges as well. Interestingly, the reference HC network that had a slightly worse BIC score, but had better relative overlap with networks from the other structure learning algorithms. With the overlap implying improved support for the existing edges and less potential overfitting, this suggests the reference network could be the more appropriate network even though its BIC score is worse.

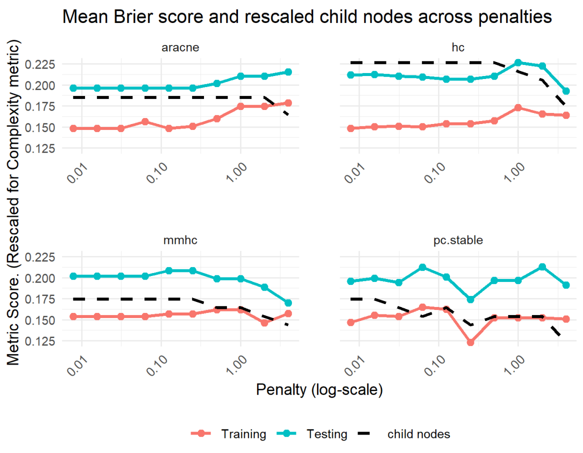
With the algorithms producing networks that perform very similarly, the Kullback-Leibler divergence (KL) is used to further compare the distributions of the fitted networks and find the candidate network that generalizes the best relative to other candidate networks. Adding up the row and column totals by network because the metric is asymmetric results in the reference HC network having the lowest total KL value. The resulting network is shown in figure 22. Similarly to the final networks discussed in the previous experimental setups, this network also has a limited amount of associations between variables from different data sources. This particular graph holds no associations between Instagram and Google Analytics variables. Interestingly, such relationships were observed in the HC network at penalty p_{-6}, and were mostly removed with increased penalization. However, the network resulting from the HC-p_0 model does contain some new associations. Most notably, the network displays associations between the two turnover variables and some causal, not co-occurring, instances of other variables. These edges, and especially the edge pointing to Turnover_decks_d7, appear in all networks except the ARACNe-HC network presented in table 11. Similar to the results found previously on the $\mathbf{X}_{T_d}$ data set, the ARACNe-HC algorithm only finds co-occurring associations which naturally excludes the edges pointing into the turnover variables, as seen in the other networks. The fact that the other algorithms find the same turnover-related edges at their respective optimal fits strengthens the support for these edges.

Comparing the networks of the PAM-based models with the Hartemink-based models, an approach identical to the one used in experimental setup X_{T_d}-B_{co} is used because the results again do not show very strong optima. Starting with network comparisons of the various algorithms, as shown in table 25, the best networks in terms of predictive performance, BIC and relative complexity for each algorithm are compared in table 12. The table shows that the models learned from the Hartemink discretized data have less overlap compared to the models in table 10, which were learned from the PAM discretized data. Moreover, increasing the penalty value did not seem to improve the (relative) overlap of the networks, meaning that the algorithms did not converge towards the same network as the reference HC network either at larger penalties. The 5 matching edges for the pc.stable and MMHC networks in the tables do coincide, but both miss 5 edges compared to the reference network. Moreover, overlap between networks from the constraint-based or hybrid algorithms appears to be very poor. This instability between algorithm outputs is particularly interesting and suggests that the data might be harder to model properly under blacklist B_{causal}. Given the $\mathbf{X}_{T_w}$ data set, this makes sense because the subsequent week aggregates are smoother by construction, which can make dependencies more ambiguous. However, the fact that this did not occur as strongly for the PAM case shown in table 10 could indicate that the PAM approach provides more stable results for smoother data.

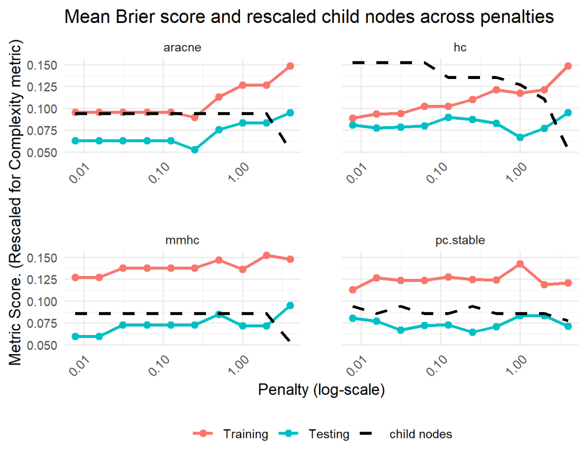
Subsequently, the KL divergence was computed for these Hartemink models. The results are displayed in table 13. Evidently, the divergence results are substantially larger compared to table 11. This implies that the probability distributions of the fitted models are more divergent. This is corroborated by the results from table 12, where it is shown that the networks have few shared edges. Although predictive performance metrics are similar between networks learned from PAM and Hartemink discretized data, the HC p_0 network learned from the PAM discretized data is the preferred model, based on its overall stability and similarity to the networks obtained from the other structure learning algorithms.

A final inspection of the arc strength plot in figure 23 for this network reveals an interesting spread. Two distinct clusters are observed at both extremes. However, there are no edges that have moderate strength levels, indicating that the associations were less vulnerable to perturbations resulting from bootstrapping. This can be explained by the fact that the larger aggregate windows provide a smoother distribution of the data.

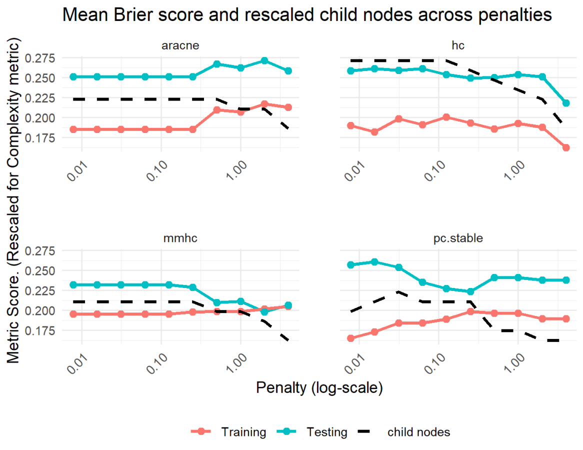
Brier score performance plots, $\mathbf{X}_{T_w}$-$B_{\mathbf{co}}$



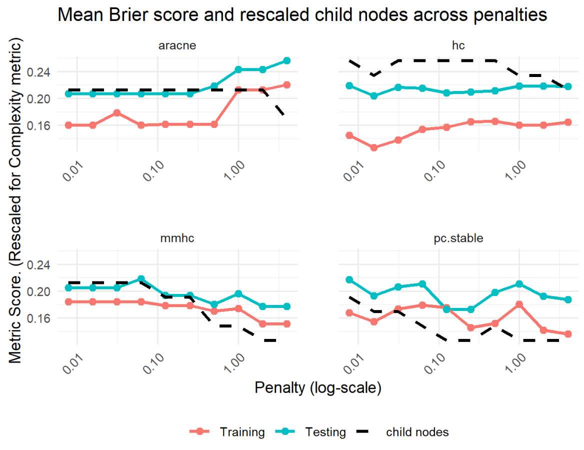
(a) PAM discretization



(b) Interval discretization



(c) Quantile discretization



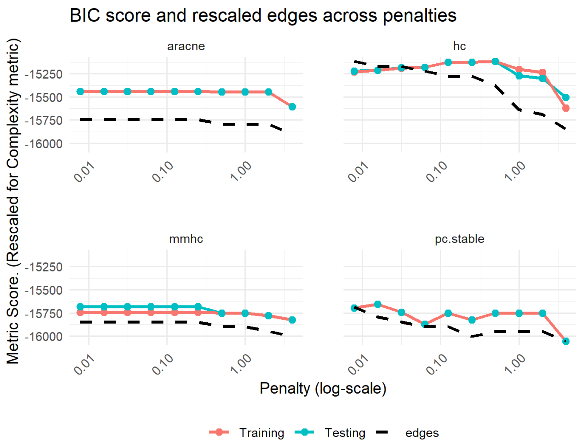
(d) Hartemink discretization

Figure 20: Mean Brier score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of child nodes present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

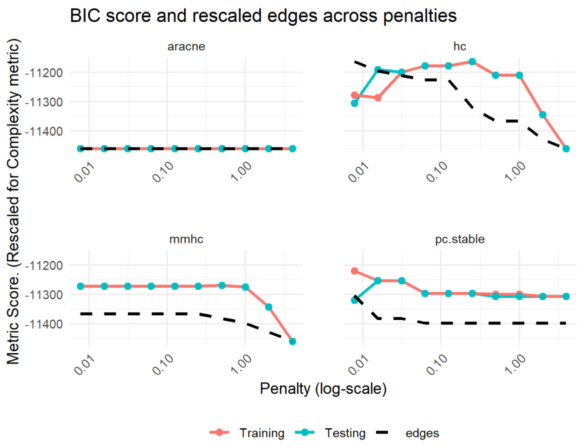
Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
HC p_{-1} vs. pc.stable p_{-7}	9	3	9
HC p_{-1} vs. ARACNe p_{-7}	9	2	9
HC p_{-1} vs. MMHC p_{-7}	7	2	9
pc.stable p_{-7} vs. ARACNe p_{-7}	6	5	6
pc.stable p_{-7} vs. MMHC p_{-7}	8	1	4
pc.stable p_{-7} vs. HC p_0	9	4	3
ARACNe p_{-7} vs. MMHC p_{-7}	6	3	5
ARACNe p_{-7} vs. HC p_0	9	2	4
MMHC p_{-7} vs. HC p_0	7	2	6

Table 10: Comparison of Bayesian networks learned from different algorithms applied to the PAM discretization of $\mathbf{X}_{T_w}$ at different penalty levels.

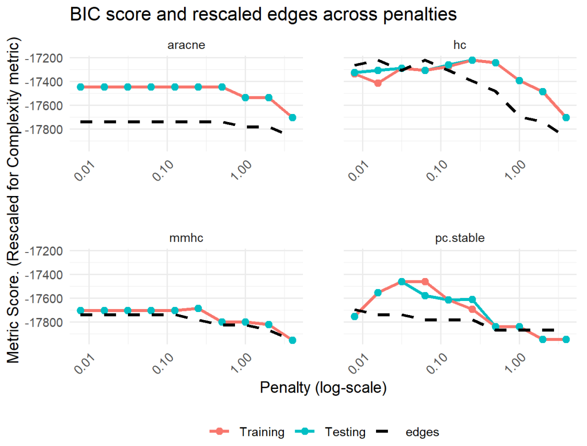
BIC score plots, $\mathbf{X}_{T_w}$-$B_{\mathbf{co}}$



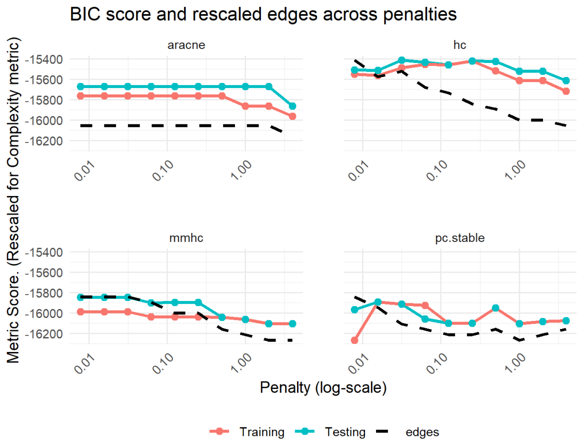
(a) PAM discretization



(b) Interval discretization



(c) Quantile discretization



(d) Hartemink discretization

Figure 21: BIC score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of edges present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

Comparison	HC p_0	pc.stable p_{-7}	ARACNe p_{-7}	MMHC p_{-7}	Row Total
HC p_0	0.000	1.593	1.735	0.813	4.141
pc.stable p_{-7}	2.265	0.000	0.818	0.818	5.387
ARACNe p_{-7}	2.685	1.261	0.000	2.711	6.658
MMHC p_{-7}	0.941	1.841	1.982	0.000	4.765
Column Total	5.891	4.696	4.536	5.8278	

Table 11: KL divergence values between fitted Bayesian network models learned from PAM discretized data, with row and column totals for comparison.

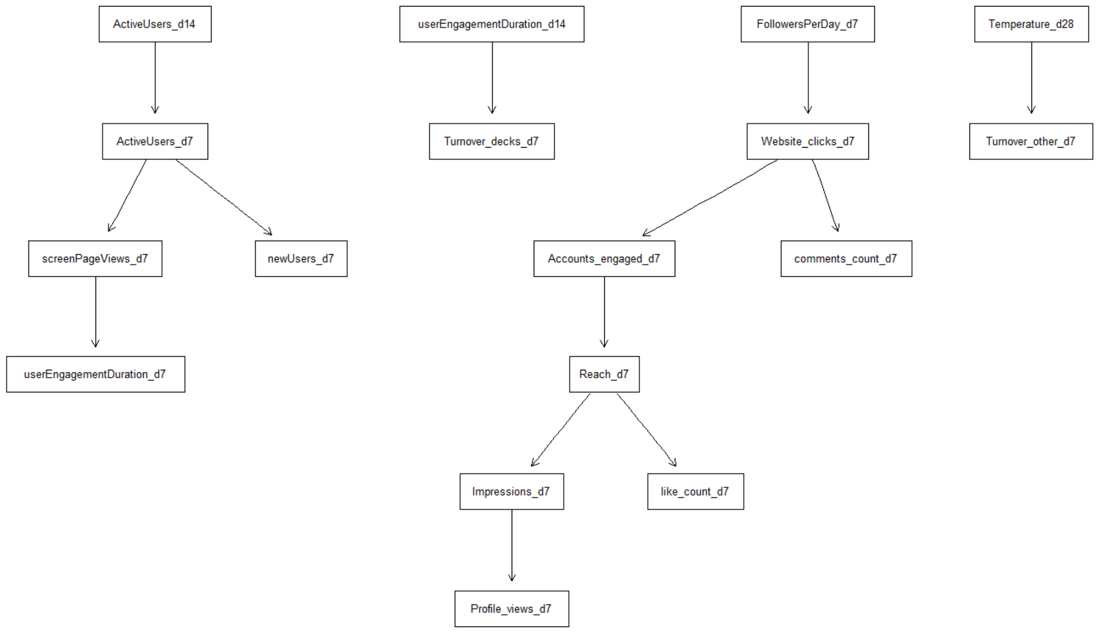


Figure 22: The network produced by the HC algorithm with penalty level p_0, learned from the data $\mathbf{X}_{T_w}$ that was subjected to PAM discretization and using blacklist B_{co}. Nodes that were assigned no edges in any direction are omitted from the graph.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
HC p_0 vs. pc.stable p_0	5	0	5
HC p_0 vs. ARACNe p_{-7}	6	3	4
HC p_0 vs. MMHC p_0	5	1	5
pc.stable p_{-6} vs. ARACNe p_1	0	9	11
pc.stable p_{-6} vs. MMHC p_{-6}	6	7	5
ARACNe p_1 vs. MMHC p_{-6}	0	9	13

Table 12: Comparison of Bayesian networks learned from different algorithms applied to the Hartemink discretization of $\mathbf{X}_{T_w}$ at different penalty levels.

Comparison	HC p_{-2}	pc.stable p_{-6}	ARACNe p_1	MMHC p_{-6}	Row Total
HC p_{-2}	0.000	3.669	4.652	2.211	10.532
pc.stable p_{-6}	4.936	0.000	1.459	3.698	10.093
MMHC p_{-6}	5.523	3.428	0.000	3.328	10.045
ARACNe p_1	2.816	3.428	3.431	0.000	9.674
Column Total	13.274	8.289	9.5463	9.237	

Table 13: KL divergence values between fitted Bayesian network models learned from Hartemink discretized data, with row and column totals for comparison.

Strength plot of HC-$\mathbf{X}_{T_w}$-$B_{\mathbf{co}}$, penalty $= p_0$

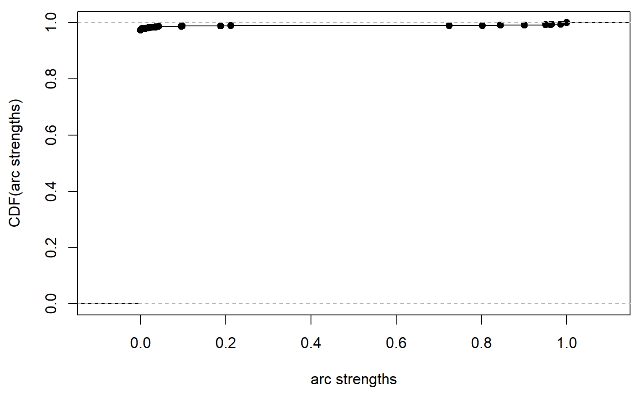


Figure 23: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge across the models obtained from all of the bootstrapped data samples.

3.5 Week-level aggregates, $B_{\mathbf{causal}}$

The final experimental setup aims to model the $\mathbf{X}_{T_w}$ using the strictly causal blacklist B_{causal}. Again, the motivation for this setup is that predicting realizations using only prior observations can be a powerful tool for forecasting if the data contain dynamic patterns that the network is able to capture. Similar to the previous results, the networks learned from the interval discretized data overperform, and the networks learned from quantile discretized data are outperformed by their Hartemink counterparts. Therefore, we will omit these plots from the figures. The mean Brier score performances of the networks are shown in figure 24. In this plot, it can be seen that the values attained by the networks contain larger numbers compared to previous figures. The fact that the mean Brier scores attain higher values directly implies that the predictive performance is worse for some models at some penalty levels, compared to previous experimental setups. This behavior is expected because the networks are constructed from a subset of edges compared to the previous experimental setup, meaning that some powerful edges could no longer be included. Most notably, figure 24b shows Brier scores tend to decrease with increasing penalties, in contrast to figure 24a, where the Brier scores seem flat or increase at high penalty levels.

Despite the low Brier scores at some penalty levels, figure 25 shows that the best fits arguably exist at low or intermediate penalty levels. Taking network complexity into consideration, the networks with very low Brier scores belong to networks that have very few edges, as can be seen in table 26. Extending the α threshold from 0.05 by doubling 0.05 over additional increments, reveals that a p-value of 0.10 results in a new structure that contains more edges and a better BIC score, but only for the networks learned from PAM discretized data, as can be seen in table 37 and 38. Another notable observation is the fact that the networks presented in tables 20 and 26 yield networks with very different complexities for the same algorithms and penalties between the two discretization methods, with Hartemink discretization resulting in networks that are closer to each other in complexity and PAM discretization resulting in networks that are either more dense or sparse in complexity.

The networks are compared within each discretization method. The results are shown in tables 14 and 15. First considering the PAM-related results from table 14, the reference HC network did not produce any strong results, so larger networks from smaller penalty values were tried as well. This did not yield any significantly better results. With regard to model complexity, it should be evident that the MMHC model in table 14 only contains three edges. This level of sparsity was maintained by the algorithm even when increasing the α threshold up to 0.2. Because the number of edges is this low, the model will not be considered a valid candidate model. However, table

14 shows that the MMHC model does represent at least a subset of a convergent structure across the networks learned from the PAM discretized data. Similarly, it appeared that such a converged network structure exists for the networks learned from the Hartemink discretized data set as well. However, these convergent structures were nearly entirely different between the two discretizations. In particular, the PAM discretization resulted in having both turnover variables belonging to the observed convergent structure, whereas Hartemink discretization resulted in autoregressive dynamic edges on the FollowersPerDay and Website_clicks variables instead.

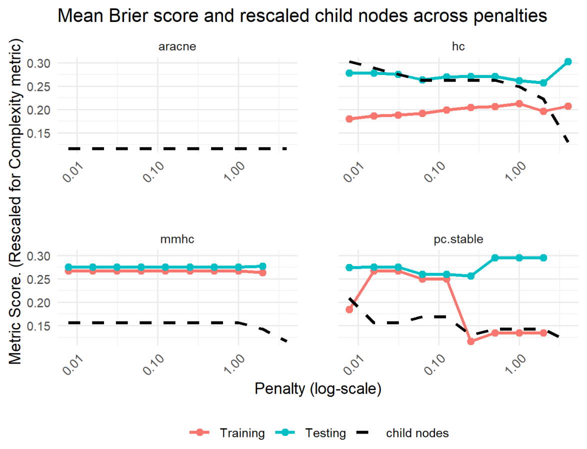
However, within their respective discretizations, the selected networks show relatively little overlap. Performing additional cross-discretization comparisons resulted in having at most five true positives when comparing the HC generated networks from both discretizations, with relative overlap being smaller than 50% for all cross-discretization comparisons. Given that the networks are quite dissimilar across the overall, the KL divergence metric can provide insight in which structure generalizes best within discretizations. The results shown in tables 16 and 17 indicate that the HC and pc.stable models are the best approximations of the other network(s) when using PAM or Hartemink discretization, respectively.

Reexamining the predictive performance of these models, it can be seen in table 20 that the HC algorithm yields a network at penalty p_{-2} that has worse predictive performance compared to the pc.stable results at penalty p_{-6} from table 26. More specifically, the model learned from the Hartemink discretized data outperforms the HC-PAM counterpart with a Brier score difference of 0.02 and precision-recall AUC difference of 0.04. Hence, we can see that the introduction of constraints, such as larger windows and extensive blacklists like B_{causal} can ultimately negatively impact the predictive performance of alternative discretizations such as PAM. Because of the interesting associations elicited by the network learned from PAM discretized data, this model will be discussed further alongside the slightly better performing Hartemink based model.

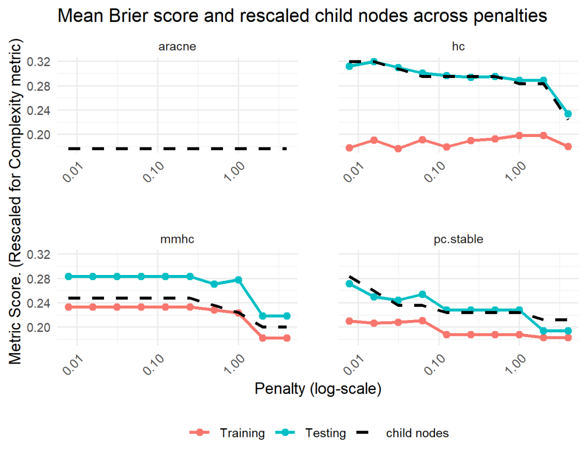
The strength plots of both models are shown in figure 26. One interesting pattern with regard to the spread of the strengths is that the model learned from PAM discretized data has a more evenly scattered spread. This implies that the edges learned for each network during the bootstrapping process are less robust, resulting in different edges appearing at intermediate relative frequencies. Hence, perturbing the distribution of variables results in different relationships between variables. In particular, the strength plot in figure 26b contains 17 edges fewer between strengths 0.25 and 0.75 compared to its counterpart in figure 26a. In order to determine whether a significant difference in edge strength distributions exists, a summary analysis is performed along with a Kolmogorov-Smirnov (KS) test. The results are shown in table 18. In addition to the edge strength lists plotted in figure 26, the best network learned from the PAM discretized data by the pc.stable algorithm is included to check if the scatter of the strength values is likely to be caused by the HC algorithm, or the discretization method. As observed in the table, it appears that the p-values returned by the KS test indicate that the PAM discretization likely produces more edges with intermediate strengths, rather than more extreme strengths. Although this represents only one instance, it hints that the more homogeneous spread of strengths is a result of the discretization method and not a shortcoming of the HC algorithm, as the same pattern is observed for another algorithm. This would imply that PAM discretization elicits more associations with intermediate strength values, with Hartemink discretization producing more extreme strength levels in contrast. Although figure 26b contains fewer edges with intermediate strengths, the edges approaching a strength of 1 seem to be more scattered in this region compared to 26a. This shows that although the spread is more extreme in general, it is less extreme for the largest strength values.

Investigating the final network in figure 27, learned from the PAM-discretized data, we can see that there are several interesting interactions at play. Namely, on the left-hand side we have the amount of active users of the previous week causally relating to the active users in the subsequent week, but also to variables from both Google Analytics and Instagram. Moreover, the number of active users during the two weeks prior to the previous week — recall that the windows do not overlap by construction— shows a stronger relationship with the amount of accounts engaged on Instagram than to the amount of active users of the previous week. This indicates that there is an effect

Brier score performance plots, $\mathbf{X}_{T_w}$-$B_{\textbf{causal}}$



(a) PAM discretization



(b) Hartemink discretization

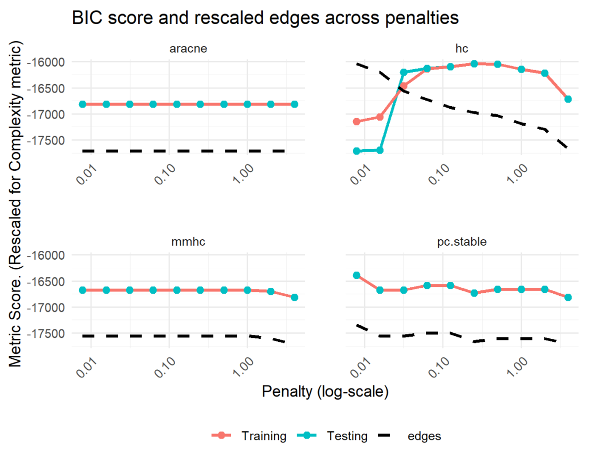
Figure 24: Mean Brier score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of child nodes present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
pc.stable p_{-9} vs. HC p_{-2}	4	10	7
pc.stable p_{-9} vs. MMHC p_0	3	0	8
HC p_{-2} vs. MMHC p_0	3	0	11

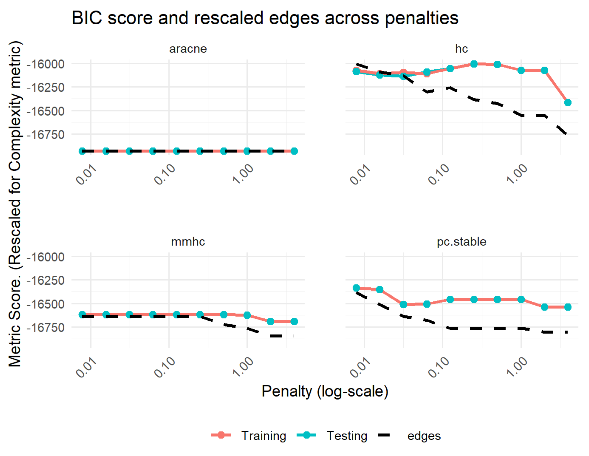
Table 14: Comparison of strictly causal Bayesian networks learned from different algorithms applied to the PAM discretization of $\mathbf{X}_{T_w}$ at different penalty levels.

at play where the amount of active users on the website translates over a longer period of time into more engagement on Instagram. Again, another cross-platform association exists between the amount of new users on the website and the total reach and impressions the Instagram posts generated that week. Again, the newUsers_d28 variable shows multiple associations that imply that new website visitors translate into more views and interactions on Instagram in the future. Perhaps more interestingly, the network suggests that both the website views and temperature have a significant co-association with the turnover of accessory items. Similarly, deck turnover is affected by the amount of time spent on the website. This association could have been expected beforehand as a co-occurring association when it is assumed that a behavioral difference between buyers and non-buyers is that buyers require additional time to place an order. However, given the fact that the parent and child nodes in the strictly causal context are aggregates of non-overlapping windows, the edge implies that there is a significant association observed between prior user engagement and the current week aggregate of sales of skating accessories.

BIC score plots, $\mathbf{X}_{T_w}$-B_{causal}



(a) PAM discretization



(b) Hartemink discretization

Figure 25: BIC score performance for each discretization method on identical train-test split samples (in terms of rows) across discretization. The dashed black line represents the relative amount of edges present in the averaged network at the corresponding penalty level. When this line aligns with the top or bottom horizontal border, it means that the maximum or minimum amount respectively across all of the algorithms and penalties is attained for that particular network.

Comparison	True Positives (TP)	False Positives (FP)	False Negatives (FN)
pc.stable p_{-6} vs. HC p_{-2}	5	8	5
pc.stable p_{-6} vs. MMHC p_{-2}	4	3	6
HC p_{-2} vs. MMHC p_{-2}	2	5	11

Table 15: Comparison of strictly causal Bayesian networks learned from different algorithms applied to the Hartemink discretization of $\mathbf{X}_{T_w}$ at different penalty levels.

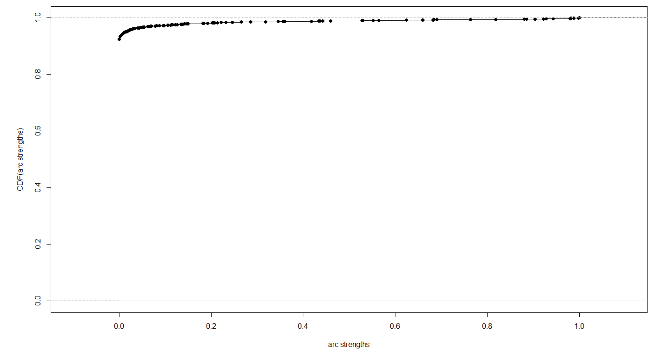
Comparison	pc.stable p_{-8}	HC p_{-2}	Row Total
pc.stable p_{-8}	0.000	5.197	5.197
HC p_{-2}	3.972	0.000	3.972
Column Total	3.972	5.197	

Table 16: KL divergence values between fitted Bayesian network models learned from PAM discretized data, with row and column totals for comparison.

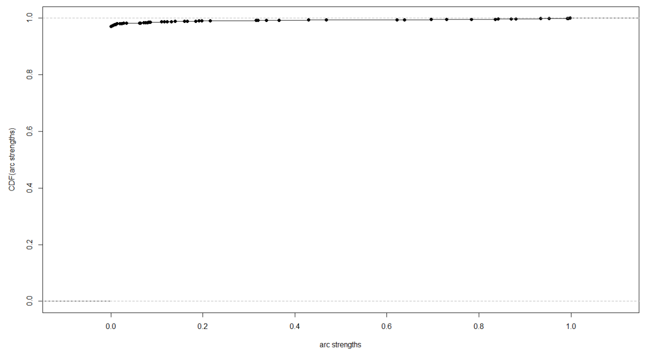
Comparison	pc.stable p_{-6}	HC p_{-2}	MMHC p_{-2}	Row Total
pc.stable p_{-6}	0.000	3.972	1.919	5.892
HC p_{-2}	2.432	0.000	3.339	5.771
MMHC p_{-2}	2.236	4.958	0.000	7.194
Column Total	4.668	8.930	5.258	

Table 17: KL divergence values between fitted Bayesian network models learned from Hartemink discretized data, with row and column totals for comparison.

Strength plots for models $\mathbf{X}_{T_w}$-$B_{\textbf{causal}}$



(a) HC p_{-2} learned from PAM discretized data



(b) pc.stable p_{-6} learned from Hartemink discretized data

Figure 26: This strength plot shows the confidence of the inclusion of edges in terms of the empirical cumulative distribution function. On the horizontal axis, we have the strength computed as the relative frequency of the edge across the models obtained from all of the bootstrapped data samples.

Model	Mean Strength	SD Strength	Weak Edges	Strong Edges	KS p-value
PAM HC p_{-2}	0.0130	0.0885	2233	14	
PAM pc.stable p_{-9}	0.0117	0.0845	2234	11	0.4675
Hartemink pc.stable p_{-6}	0.0074	0.0845	2242	11	0.0181

Table 18: Comparison of edge strength mean and standard deviation statistics for different discretization methods and structure learning algorithms. The p-value presented represents the p-value returned by the KS test between the model in the first row and the respective row.

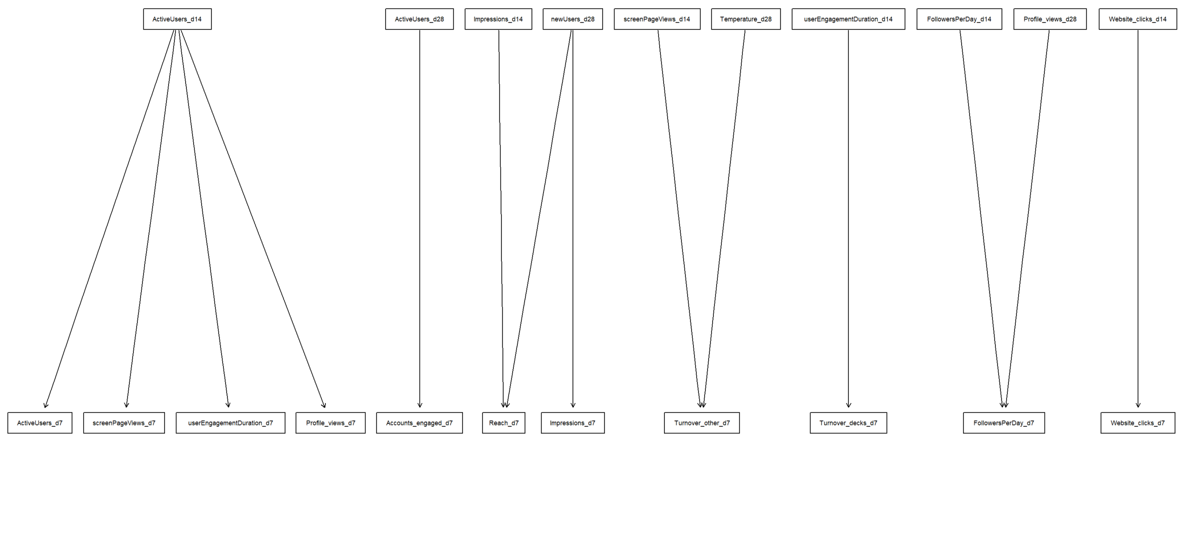


Figure 27: The network produced by the HC algorithm with penalty level p_{-2}, learned from the data $\mathbf{X}_{T_w}$ that was subjected to PAM discretization and using blacklist B_{causal}. Nodes that were assigned no edges in any direction are omitted from the graph.

3.6 Comparison of bin usage

A final investigation was performed on the impact of predictive performance when increasing the number of bins used in the discretization step of processing the data. In addition to the use of three bins, networks were also learned from data sets that were created when using four, five or three to five bins were used. At this stage of the research, the quantile and interval discretization methods were omitted because of the results that were previously found. However, attempting to perform Hartemink discretization on the data set $\mathbf{X}_{T_d}$ while imposing more than three bins would result in empty bins for several variables. Therefore, only results from networks learned from $\mathbf{X}_{T_w}$ will be presented when discussing the results of the Hartemink discretization. In contrast, results for both $\mathbf{X}_{T_d}$ and $\mathbf{X}_{T_w}$ were found for all numbers of bins imposed when using the PAM approach.

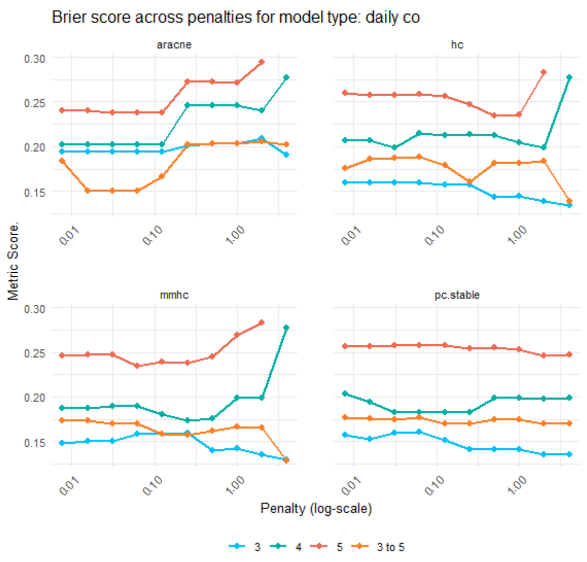
The results of the Brier scores of the networks learned from $\mathbf{X}_{T_w}$ and $\mathbf{X}_{T_d}$ using PAM discretization are shown in figure 28. Using the expected increase in the Brier score from equation (31), the results in this figure can be interpreted. Increasing the number of bins from three to four leads to an expected increase in Brier score of $\frac{1}{12} \approx 0.08333$. Similarly, increasing the number of bins from four to five comes with an expected increase of $\frac{1}{20} = 0.05$. Lastly, in the case of the mixed bins, where the amount of bins was determined according to equation (32), the weighted average was used proportional to the amount of variables for each number of bins greater than three. For $\mathbf{X}_{T_d}$, 27, 20 and 17 variables were assigned three, four and five bins respectively, resulting in an expected increase in Brier score of roughly 0.061. Similarly, 21, 15 and 12 variables were assigned three, four and five bins respectively for $\mathbf{X}_{T_w}$, resulting in an expected increase of Brier score of roughly 0.059.

Figures 28a and 28b show most structure learning algorithms display expected behavior. In these figures, the increase in bins appears to consistently result in an increased value of the Brier score across penalties, with only some deviations in behavior at the extreme end of the penalties where the networks have become very sparse. Using the expected increase in Brier score, to contextualize the observed increases, the mixed bin number strategy seems to often result in an increase that is smaller than 0.061 compared to the default three-bin strategy for nearly all penalty values across algorithms. In contrast, the increase in bin number from four to five is less powerful in the sense that its increase in Brier score is sometimes greater than 0.05.

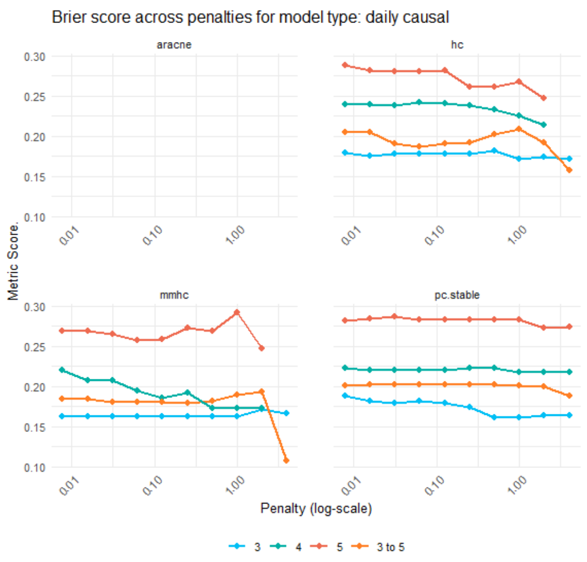
Considering figures 28c and 28d, the results are less consistent. The HC algorithm appears to result in clear differences between the different bin numbers. The ARACNe algorithm shows consistent behavior for each bin, but the resulting networks appear to suffer disproportionally from increasing the number of bins above three relative to the expected increase in Brier score. Lastly, the MMHC and pc.stable algorithm results elicit somewhat erratic behavior. More specifically, these algorithms show that increasing the number of bins sometimes results in favorable Brier scores. This is observed most clearly from the mixed bin results showing some of the worst performances for these algorithms, as well as the five-bin discretizations outperforming the four-bin discretization in the co-occurring case.

The results for increasing the bin number when using the Hartemink discretization are shown in figure 29. These figures provide a more concise effect of increasing the bin number. It appears that using five bins is in some cases not favorable, as the increase in Brier score is very large compared to the observed Brier scores observed when using four bins. These cases are most salient in weekly causal setup of figure 29b and occur only for the MMHC and pc.stable algorithm in figure 29a. Only in a few cases does having four bins create a competitive model in terms of Brier score. In these cases, with the best example being the pc.stable algorithm results in 29b, the Brier score increase is not only noticeably smaller than 0.0833, but even attains very similar Brier scores between networks learned from three and four bins. No mixed bin strategy is displayed here because the mixed binning approach for Hartemink discretization resulted in assigning five bins for every variable.

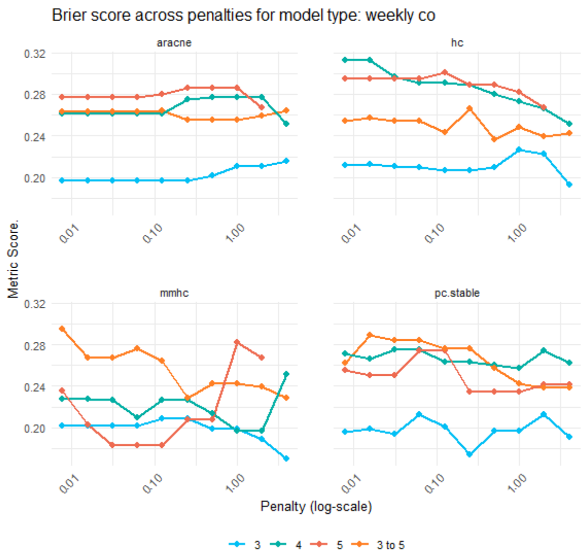
Brier score plots using different bin numbers under PAM

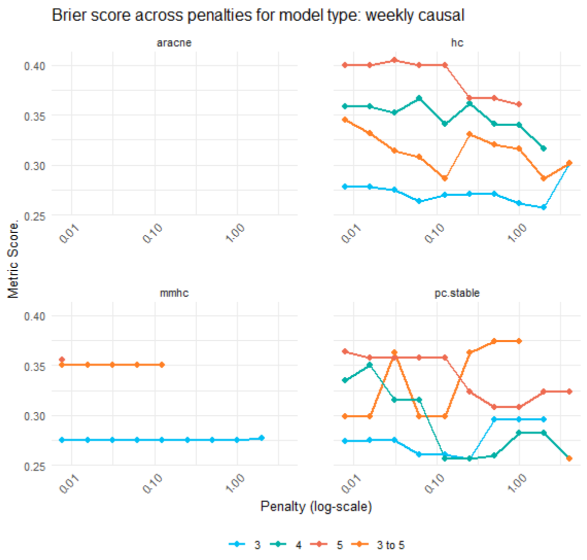


(a) Brier scores for networks learned under $\mathbf{X}_{T_d}$-B_{co}



(b) Brier scores for networks learned under $\mathbf{X}_{T_d}$-B_{causal}

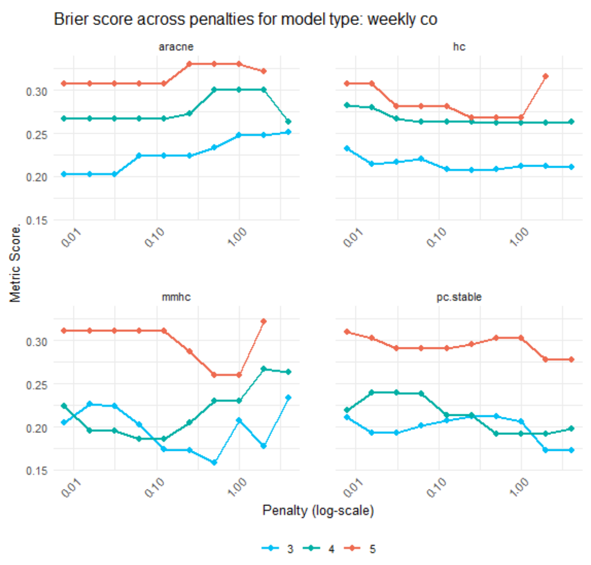




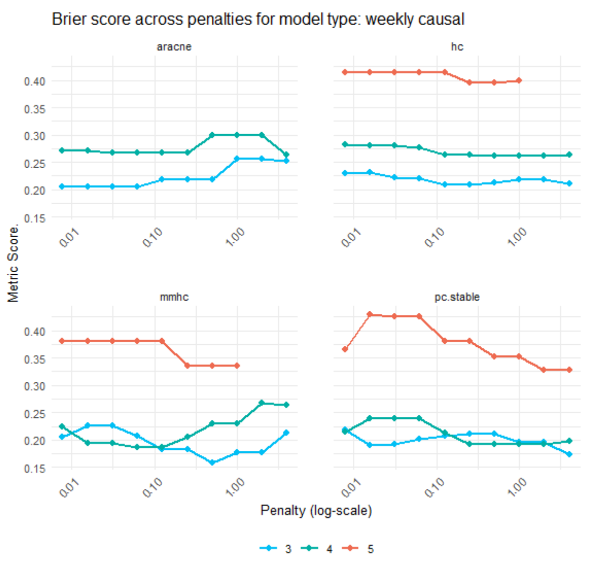
(c) Brier scores for networks learned under $\mathbf{X}_{T_w}$-B_{co} (d) Brier scores for networks learned under $\mathbf{X}_{T_w}$-B_{causal}

Figure 28: These four plots present the predictive performance in terms of Brier score for the networks learned from their respective data set subjected to different levels of binning. Each line represents some amount of bins that was applied during the discretization process, with the exact amount shown in the legends of the graph. Note that in this figure each of the four sub-figures represents a particular experimental setup regarding the aggregation of the variables and the blacklist that was used, rather than a discretization method.

Brier score plots using different bin numbers under Hartemink



(a) Brier scores for networks learned under $\mathbf{X}_{T_w}$-B_{co}



(b) Brier scores for networks learned under $\mathbf{X}_{T_w}$-B_{causal}

Figure 29: These four plots present the predictive performance in terms of Brier score for the networks learned from their respective data set subjected to different levels of binning. Each line represents some amount of bins that was applied during the discretization process, with the exact amount shown in the legends of the graph. Note that in this figure each of the four sub-figures represent a particular experimental setup regarding the aggregation of the variables and the blacklist that was used, rather than a discretization method.

4 Discussion

One of the recurring issues encountered in this research is the performance of both interval and quantile discretization. In the case of quantile discretization, the fact that these networks were outperformed by networks learned from Hartemink discretization is not surprising given that Hartemink discretization is essentially designed to optimize overall patterns in the data (Hartemink, 2001). Nevertheless, these networks learned from the quantile-discretized data displayed normal behavior but simply performed worse. This highlighted one of the main benefits of using quantile or Hartemink discretization. That is, these discretization methods result in well-populated bins. In the case of quantile discretization, this is obvious, but this is less so the case for Hartemink discretization. Because the Hartemink approach is based on pairwise mutual information, the bin sizes are naturally relatively similar, especially for smaller amounts of bins. In practice, such bins produce stable results because bins generally are decently populated.

In contrast, the networks learned from interval discretization displayed worrisome behavior. The networks consistently performed better on the test set than on the training set. Moreover, both the training and test performance metrics were extremely high. This indicated that this discretization method likely produced bins that were not specific enough. Upon more detailed inspection of the resulting data set, it appeared that the vast majority of observations were assigned the value 1, corresponding to the bin representing the lowest values. Because Custom Decks experiences heavy fluctuations in metrics because of posts going viral and sudden peaks in orders among other things, virtually all baseline activity got assigned to this bin. In addition, because of these peaks, the second bin contained mostly values that would lead to the optimum of some peak and contained barely any other observations. Therefore, the labels 2 and 3 were underrepresented but, most importantly, not very informative. Consequently, this was also reflected in the predictive performance. Namely, the Brier score and accuracy metrics were very high for the networks learned from this discretization, but the precision-recall AUC value was moderate or comparable to networks from other discretizations. This finding implies that the majority classes are often properly predicted, but perform weaker on rare observations from the minority class. All in all, this meant that this discretization was unusable for the data at hand and all other methods were preferable in comparison.

Considering the fact that the goal is to ultimately query the network and have meaningful results, the discretization method used had to bear some meaningful interpretations. On this front, the quantile discretization employs a less meaningful assignment of labels that does not strongly discern informative patterns in the data and was therefore not preferable. The Hartemink discretization implicitly holds a lot of information because it optimizes the pairwise mutual information. However, because the computation of these labels under this method is quite extensive in the sense that many variables are involved and influence the label assignment, the interpretation of the proposed bins can become quite convoluted. Therefore, the distance-based algorithm, PAM, was introduced to provide an informative, yet reasonably interpretable binning structure. Upon inspection of the networks learned from this discretization, it appeared that the performance was quite similar. That is, the networks learned from PAM discretized data boasted similar predictive performances in terms of Brier score, accuracy and precision-recall AUC. However, in some levels of penalization, the networks performed similar or even slightly better on the testing data set compared to the training data set. This behavior warrants caution because it might indicate that the test sample is too simplistic, similar to the behavior that was observed in interval discretization. It appeared that this behavior was rooted in the type of sampling that was used to generate the testing sample. That is, because this sample was created by taking a tail part of the data set, rather than a random sample, particular trends that recently appeared in the data would only be captured in this sample. In the case at hand, this means that a more quiet sales period would naturally result in more observations belonging to the first or second bins for many variables. This is more apparent when using PAM discretization compared to Hartemink discretization because PAM captures local clusters within a single variable more strongly than Hartemink, which tends to produce more evenly populated bins in comparison. Similar to interval discretization, this might lead to sparsely populated bins. This explained the behavior of seemingly overperforming predictive metrics in some instances of the PAM algorithm. However, it could be argued that because of spatially motivated clustering with PAM, this behavior is acceptable. Especially in context of the precision-recall AUC values being similar between networks learned from PAM and Hartemink discretizations for the selected models, this implies that the overall performance of the PAM does not display any major concerns. Another recurring issue was the ARACNe algorithm not returning any edges when using the causal blacklist B_{causal}. This behavior appeared to originate from the fact dynamic, strictly causal associations are typically weaker compared to co-occurring associations. Because the ARACNe algorithm prunes edges that have comparatively pairwise inferior information, such strictly dynamical edges were more likely to be beaten by co-occurring edges. This, combined with the conditional independence test, G^2, used in the ARACNe algorithm appeared to result in co-occurring edges dominating the network. However, even when blacklisting co-occurring edges, no edges would appear, indicating that a p-value of 0.05 would not be matched by the conditional independence test when using strictly causal relations.

One of the core challenges in this research is that it is very specific to Custom Decks' data and that Bayesian networks have barely been applied in this e-commerce context. In addition, no true models were known beforehand to validate the learned networks. To tackle this issue, a benchmark data set was used to identify at which algorithm would produce the most appropriate network. This provided a point of reference to use for model comparison. If expert knowledge had been applied through whitelisting, it could have served as a minimal backbone to learn and build the potential remaining structure around. However, because this endeavor is very specific to a business, this was omitted from this project's general approach to protect generalizability.

Consequently, networks were learned from various starting points (depending on the algorithm) with blacklists and penalization factors as input. The resulting networks had varying complexities, where the HC algorithm often returned networks with largest number of nodes and edges. The pc.stable algorithm typically produced networks that had a moderate amount of edges, that would decrease with the amount of penalization. In contrast to the completely constraint-based pc.stable algorithm, the ARACNe-HC and MMHC algorithm would learn structures using constraint-based structure learning and would further optimize within the space of this structure with Hill Climbing. However, the structures learned in the first part of the hybridized algorithms would often be relatively small to begin with, and therefore very robust to increasing penalization. As visibly apparent in the Brier and BIC plots, both the ARACNe-HC and MMHC algorithms would often return networks with identical networks across most of the penalty levels. Because the α threshold was scaled based on the penalty value —penalty p_{-7} would correspond to a p-value of 0.05 that was halved upon each subsequent penalty level—, these algorithms would appear to only reduce in complexity when extreme values for α were imposed for the conditional independence tests. Interestingly, the BIC scores would always decrease when edges were finally removed, showing that these structures could not be improved by pruning.

Ultimately, most of the presented networks had intermediate complexities, with edge counts ranging between 10 and 15. We would like to note that in reality this amount of edges is low compared to the amount of nodes available in the network, and that the term 'intermediate' is therefore used to describe the size of the edge list relative to the minimum and maximum observed numbers of edges in the networks generated at the initially proposed penalty levels. Only in the latter experimental setup was there a case where the initially chosen penalty range was not sufficient. This was the result of the pc.stable algorithm returning a denser network with a better BIC score when a p-value of 0.20 was imposed. Although increasing the α threshold above 0.05 is generally not standard practice because it allows for more coincidentally appearing associations to be introduced, it did result in a network with said intermediate range of edges, as was seen in table 20. Although the overall performance of this network did not beat the selected HC model, its predictive performance metrics were only slightly worse than its counterpart, the pc.stable network, that was obtained at the regular $\alpha = 0.05$. Given the fact that the complexity of the network increased by more than 50% when increasing the p-value and the newly included edges posed weaker associations, this finding was all the more interesting given its better BIC score and its comparable predictive performance. Although this case was ultimately too weak to beat other models and is rare to begin with, future applications in this context might benefit from having a range of α thresholds that extend beyond 0.05 and below.

Returning to the discussion on the desired complexity of the final networks, several characteristics were considered. As presented in the results, some networks from different algorithms would converge to highly similar structures when heavy penalization was used. However, structural differences were observed in the networks learned from different algorithms. Table 12 showed a particularly remarkable result with the MMHC and ARACNe-HC networks having only few or even zero edges in common with the networks from the other algorithms. This finding highlights the potential volatility in structure learning, even when using Hartemink discretization. Such divergent structures were achieved with penalties that yielded the best BIC score, meaning that these models achieved different local optima. Unfortunately, there is no obvious solution to determining the preferred network or optimum. In practice, one could resort to assigning weights to nodes or even edges to reflect their importance. Using this method, the networks could be scored based on a cumulative sum of weights of the nodes or edges in the network. Alternatively, the weights could be integrated in the scoring functions used inside of algorithms to steer the algorithms towards opting for more important associations.

Ultimately, applications like these where domain knowledge is not readily available because of the novelty of the application make it hard to know when a network is truly suitable. This makes a true network rather elusive. Even with scoring functions, there is no structural way to validate whether a network accurately reflects all patterns in the data, not to mention the potential presence of latent variables. Nevertheless, application of the structure learning process to the Hailfinder benchmark data set in the preliminary analysis section, provided a reference point for finding plausible networks. This resulted in the HC algorithm providing the best results. Naturally, this meant that the reference HC network was defined as the network learned from the HC algorithm at penalty level p_0. With the true network being unknown, there is also no indication of the proper amount of associations that accurately describe all of the significant patterns in the data. Here, the reference HC networks provided a means of comparing the results of other structure learning algorithms in a meaningful way. Given the fact that different algorithms and discretizations resulted in different networks, where convergent results did not appear structurally within each algorithm or discretization for each experimental setup, many candidate edges from different networks were effectively rejected in the final network. This wide-ranging collection of associations that were proposed among all the networks could indicate that the true network could be more complex than the networks that were ultimately selected in their respective experimental setup. If this is true, the algorithms used in search for this true network were unable to find this network, and other algorithmic approaches would be warranted.

Another interesting comparison between the preliminary analysis and the main results was the difference in network density. In particular, the network densities of the initial networks learned without model averaging and default penalty p_0, shown in figure 6, were much higher than the densities found in the averaged networks. Moreover, the averaged networks had multiple subgraphs, which were not observed in the initial networks from the preliminary analysis. This behavior of the averaged networks was consistent between configurations in terms of blacklists and variable aggregates. Perhaps the most notable part was that the averaged networks provided more overlap in comparison to the non-averaged networks. This showed that the practice of model averaging in combination with bootstrapping can boost the stability of the results and prevent overfitting to the original data.

Another interesting finding about the different experimental setups is the nature of the associations that are uncovered by the various experimental setups. Most importantly, the type of blacklist plays a major role in influencing the nature of associations. However, even when using blacklist B_{co}, most algorithms would select strictly causal associations at some point in the structure learning process. Interestingly, the hybridized ARACNe-HC algorithm would always exclude such causal associations. Focusing on the networks produced by the algorithms that did include these dynamical edges, either between different aggregates of the variable itself or between different variables, there are some noteworthy observations to be made. Starting with the dynamical relationships found in the first two experimental setups that used delay set T_d, the pc.stable, HC and MMHC networks contained some dynamical edges. Most of these edges had an autoregressive nature, as can be seen in figure 9. Here, only the like_count_d2 variable contains an edge that is not of this nature. In addition, it should be mentioned that the X_{T_d} data set contained single-day aggregates _d1 and _d2 as well as week aggregates _d7 and _d14, but none of the nodes representing these weekly aggregates were assigned edges at the end of the structure learning and model averaging processes used. Given that all dynamic relationships were given by _d2 variables, it implies that not a single week aggregate posed associations strong enough to outperform these edges, nor provide statistically significant additional associations to become a co-parent.

In contrast, the delay set T_w resulted in different relationships in general. Comparing figures 9 and 22, there are several observations to be made about their respective structures. First of all, the edges in figure 9 are more complex, with fewer but more cluttered subgraphs. Perhaps more interestingly, the dynamic associations implied by some edges were never autoregressive in the final network presented for the weekly case, as seen in figure 22. Moreover, the associations implied by these edges were cross-platform, providing some more interesting insights. This behavior was hinted at by the results from the cross-correlation matrices in figures 3 and 4. Similarly, the strictly causal network shown in figure 27 showed relatively few autoregressive edges. Although autoregressive associations are generally plausible to observe in networks learned under blacklist B_{causal}, these edges did not appear to dominate non-autoregressive dynamical edges. This implies that different and more complex interactions might be at play between variables when they are aggregated to week scales. Such cross-variable associations are valuable to provide more diverse and actionable associations that can be used in business strategies. The different nature of the associations that appear for these different delay sets indicates that these delay sets have a notable influence on the resulting network. As such, this means that different patterns are elicited when using different aggregation windows, which can be leveraged for different purposes in practice.

An important nuance to add to the effects of the delay sets is with respect to the different discretizations and structure learning algorithms that were used. In addition to consulting the tables that compared network edge overlap, the networks compared in these tables were visually inspected, as well to check for interesting structures. Throughout these inspections, it became clear as well that the overall nature of the edges was strongly consistent within a particular delay set. That is, the presence of mostly autoregressive or more cross-variable associations seemed to be stable across discretizations methods and structure learning algorithms within a particular delay set. This finding corroborates the previous observation that the PAM and Hartemink discretizations do not tend to model fundamentally different types of interactions. Although the exact edges selected might differ to some extent, as seen in the comparison tables, these methods tend to use similar types of edges.

This variation in the edges found in the networks between different discretizations is a challenging aspect of this study. It can be easily seen that imposing different binning strategies on the same data set will likely produce different associations at some point. In case the data sets resulting from the different binning strategies are similar, the networks learned from these data sets should be similar as well. However, given the different natures of Hartemink and PAM discretization, the interpretations of the networks should arguably be separate. A visual example of this was shown in figure 1, where the Hartemink discretization resulted in a much wider bin for the larger values because these values would occur less frequently in the data. In contrast, the PAM discretization provided bins of a more similar width. These clear and distinctive differences can occur for any variable in the resulting discretized data sets, meaning that direct comparison of networks between these discretization types should be treated with caution. Therefore, in cases where the discretized data sets differ, it is warranted to interpret these networks as modeling different patterns in the data.

Combining the latter two points, some interesting insights can be obtained. First of all, it appears that the types of edges observed are relatively stable between discretization methods. Secondly, the bins resulting from the different discretization strategies can differ strongly. Therefore, the fact that both of these observations hold, implies that both strategies have similar merit, but reveal different patterns. Assuming that the interpretation of the networks rests on the nature of the bins, one could argue that these discretization strategies can or should be used to reveal fundamentally different insights.

Another interesting finding was the robustness and similarity of networks for the structure learning algorithms within discretization methods. In particular, with the use of $\mathbf{X}_{T_w}$ it appeared that the PAM discretization resulted in networks that were similar between structure learning algorithms, in contrast to Hartemink discretization. As mentioned, this is likely rooted in the smoother data observed in $\mathbf{X}_{T_w}$ (compared to $\mathbf{X}_{T_d}$) that makes the bin counts in the PAM discretization more balanced. However, because increased smoothness also makes dependencies more ambiguous, it reduces the power of the mutual information-based Hartemink discretization. Under this assumption, it seems plausible that the resulting less informative bins make the networks from different structure learning algorithms less likely to be similar. Vice versa, this line of reasoning would imply that Hartemink discretization would be more appropriate when discretizing $\mathbf{X}_{T_d}$, given that it has stronger dependencies that the Hartemink algorithm can leverage, and makes bin counts under PAM discretization more imbalanced.

An interesting takeaway on the similarity of the networks within the PAM discretization of $\mathbf{X}_{T_w}$, is its degree of reproducibility. It was seen for strictly causal networks that similarity of networks was obtained for neither PAM nor Hartemink discretization. Given that this similarity between networks was found when using the PAM-discretized $\mathbf{X}_{T_w}$ and B_{co}, but not when using B_{causal}, implies that such robustness of the network structure between structure learning algorithms can depend on the blacklist. The causal networks require the use of a subset of edges and can therefore be prone to relying on weaker associations, hence introducing more variance in the graph structure. Under these conditions, it could be argued that the reference HC network should be the default network to be used. Still, this highlights that the data used might not provide enough support for robust causal dependencies. In addition, this could be due to the limited number of observations. Hence, using a larger data set could help determine if this instability is caused by weak associations rather than sample size.

The final part of the results presented the impact of the number of bins used in the data when learning a network. A significant amount of results showed that the increase in the Brier score was lower than the expected amount. This implies that increasing the number of bins provides additional information. Still, the problem becomes more challenging when using more bins, which naturally explains the somewhat inevitable increase in the Brier score. The cases where the observed increase was similar to the expected one indicate that no additional information was obtained from the introduction of new bins. Therefore, the cases where an increase in Brier score was lower than expected showed that introduction of additional bins was beneficial. By carefully inspecting which algorithms provide worthwhile results under the addition of bins, the network(s) ultimately used in their respective applications will benefit from an enhanced representation of the data, compared to using only a fixed number of bins.

One notable observation about the behavior of the networks was their robustness when the number of bins increased. Especially with the use of PAM, the predictive performance of the networks across bin numbers appeared to be heavily influenced by the nature of the data. In particular, using the $\mathbf{X}_{T_d}$ data resulted in more stable results compared to the results of the networks learned from $\mathbf{X}_{T_w}$. The suspected cause of this is the presence of many short-term causal edges that were present in these models, as discussed above. These edges naturally have a stronger association, as hinted in the preliminary analysis, which, in turn, makes them more robust to complicating the classification problem by introducing additional bins. In the same vein, the weekly aggregates had weaker cross-correlations and are therefore most likely less robust, complicating the classification problem. Furthermore, the results obtained from the $\mathbf{X}_{T_w}$ data were also less consistent, in the sense that the number of bins did not always reflect a proportional increase of the Brier score.

4.1 Future recommendations

One of the core components used in the various experimental setups was the level of aggregation of the time lags used. Using daily and weekly aggregates was done based on insights retrieved from the cross-correlations observed between a wide range of aggregates. Given that the results are not extremely consistent within some experimental setups, it would be worth exploring if aggregates of 3 to 6 days could provide a middle ground for relatively strong dependencies, while also utilizing larger aggregates that can be more useful in real-world applications. Similarly, the use of even larger aggregates, such as up to two to four weeks, as the smallest aggregates in the variable list might be worthwhile. This could have some merit given the fact that figure 5 shows an increasing amount of mean absolute cross-correlation as the aggregates become larger. Such larger aggregates produce a smoother data set, meaning that a larger aggregation strategy could be worthwhile when using PAM discretization in particular. Hence, there could still be merit to exploring more ranges of aggregations. Another weakness of this study is the lack of a large data set. Although there were hundreds of observations, some bin counts could become sparse, especially under smaller delay aggregates. This makes it unlikely that true but weak dependencies are identified in the structure learning process. Having a larger data set would therefore help to produce more interesting networks, as well as more stable results.

In addition, the approach taken here was strictly for discovering dependencies, while assuming no prior network based on expert knowledge. However, several edges could have been included that should be present by definition due to certain known business processes. This could have been accomplished by using a whitelist. These whitelisted edges could have helped include edges that were otherwise omitted because of the lack of support due to the sample size in the given data set. Moreover, including edges through whitelisting could have pushed structure learning processes in directions that would produce more convergent structures between structure learning algorithms. Therefore, investigating the list of candidate edges for the whitelist and applying this list could also improve overall results.

The current study used a total of four different structure learning algorithms. Although these represented a variety of structure learning methods, the fact that these different algorithms displayed different behaviors from each other quite consistently within each experimental setup suggests that exploring a wider variety of algorithms could reveal more suitable candidates. For example, the ARACNe algorithm showed that it was not well suited for the weaker associations found when using a strictly causal edge-imposing blacklist. Combining this with a more liberal range of α values for the conditional independence tests could result in finding more viable structure learning algorithms.

An issue that was not highlighted in this study was the potential presence of latent variables. Simply put, there is no knowing whether the current set of variables would have suffered from latent variables. Although more variables could have been collected from the APIs to combat this, there is no guarantee that no latent variables would remain. In extreme cases, dimensionality reduction methods could help prune collinear variables prior to structure learning. Another approach could be to manually add a latent variable that becomes the parent of a cluster of nodes that are highly correlated (Koller & Friedman, 2009). By blacklisting any other edges from connecting to these correlated child nodes, nodes outside of this cluster only form dependencies with the new latent variable that represents the construct that the cluster of child nodes share. This simplifies the network in general and prevents the correlated nodes within the cluster from forming spurious edges. These methods, among several others, can improve the validity of the data or network at hand and should hence be considered in general. Future research on dealing with latent variables in business- or sales-related data in the context of Bayesian networks is therefore recommended.

Lastly, the number of bins used in all but the last section of the results was three. The choice of this number was because not all variables would permit any number of bins with Hartemink discretization. In order to compare the resulting networks from different binning strategies, this study opted for minimizing the differences in binning numbers between discretization methods, which meant that the data were discretized into three bins for all algorithms for consistency reasons. Especially in the context of PAM discretization, this was practical because it would divide the data into low, medium and high values in a natural way. However, because the addition of more bins resulted in promising results in terms of predictive performance, it would be worthwhile to inspect the curve of the loss of information per bin number when using Hartemink discretization so that an informed bin number can be used. This would likely result in a higher bin number being recommended. Although this would make predictive performance worse by definition because there are more options, it would increase the amount of information the structure learning algorithm can leverage to assess dependencies, as shown in Hartemink (2001).

5 Conclusion

The aim of the thesis was to answer three research questions. These questions regarded the suitability of partitioning around medoids (PAM) discretization compared to the more conventional Hartemink discretization, the suitability of certain structure learning methods and the impact of different types of blacklists for dynamic networks. Each of these questions covered a vital aspect of applying Bayesian networks in a business setting so that the network would be useful and informative. Moreover, different scopes of time aggregates were used to create different versions of the data. This was done to explore the how different temporal aggregates affect network structures, but also because different scopes would be of interest to a business. The combinations of discretization, structure learning algorithm, blacklist type and data aggregation resulted in a wide variety of models for which the results were analyzed extensively. Ultimately, different results were found between the different configurations of the models and corresponding data preparations.

From the perspective of discretization, the PAM method had interesting results. It was included in the research because of its more intuitive representation of a single variable, compared to Hartemink discretization. It appeared that the results depended to some degree on the way the data were aggregated, with larger, and thus smoother aggregates providing more stable results compared to Hartemink discretization. In the case of short-term daily aggregates, similarly stable results were found when using either discretization method. Both methods had trouble providing stable results when modeling the most difficult configurations regarding larger aggregates that come with weaker associations, in combination with a causal blacklist. This underscored the limits of the methods when the body of data is not very large. All in all, with regard to learning networks, the PAM algorithm showed clear superiority over the likes of interval and quantile discretization, and was competitive with the Hartemink method.

Secondly, four structure learning algorithms were utilized when learning networks. These algorithms were either score-based or hybridized constraint-based methods. It appeared that when constraint-based methods were used, the statistical significance tests for assessing dependencies were often quite limiting. That is, within the local search space imposed by a constraint-based algorithm, only so many candidates would ever pass the α value imposed by the penalty level used. This resulted in these algorithms providing sparser networks, or networks that would almost be invariant under different levels of penalization. Only when the threshold α for these tests was relaxed, some denser networks would be found.

In addition, the score-based HC algorithm provided results that appeared to rely more strongly on the level of penalization. However, these were included for comparison purposes mostly, because the network found at the standard penalty level p_0 would be the reference network, based on the preliminary analysis. Across the experimental setups and discretization methods, the reference networks would in most cases be highly similar to many other networks. This showed that these networks had clear merit and could potentially be used in practice with reasonable levels of certainty about their validity.

Third, the type of blacklist appeared to have a high impact in general. Namely, when using either daily or weekly aggregates to represent the data, using a causal blacklist caused the models to lose predictive performance compared to using the co-occurring blacklist. This was expected because the causal blacklist is a subset of the co-occurring blacklist. However, the use of the causal blacklist also reduced the stability of the results. Although the networks learned from daily aggregated data were affected by this to a lesser degree, both versions of the data sets would result in greater variance between the networks learned from the different structure learning algorithms. This showed that such dynamic models might be less appropriate for the data at hand. Hence, either more observations or more informative data would be needed to improve the stability and reliability of these networks.

Lastly, the use of larger numbers of bins was explored. The results showed that using more bins could prove worthwhile because the corresponding increase in Brier score was lower than the expected increase under random guess. Future applications could benefit from performing a preliminary analysis on the mutual information retained by the Hartemink discretization and the number of medoid clusters that minimizes the overall distances within clusters.

There remain several promising avenues for future work. Much potential lies in the use of intermediate or very large windowed aggregation of the original data set, in contrast to either single-day or week aggregates as the child nodes. Additionally, including a domain knowledge-based whitelist could help push networks to more convergent structures across structure learning algorithms. Working with a larger data set could remedy some of the issues described above or encountered during the analysis. In combination with employing more structure learning methods as in Scutari, Graafland, and Gutiérrez (n.d.), these recommendations could help uncover even more refined insight about how to properly model time-series data found in real-life business applications.

In conclusion, the different discretization methods, data set aggregates, and blacklists provided much insight into the applicability of Bayesian networks in a business setting. It highlighted the trade-offs between different structure learning methods and the comparability between networks learned from differently discretized data. Also, the use of causal blacklists showed the vulnerabilities of the various structure learning algorithms and the caution required when working with such more heavily restricted model configurations.

6 Appendix A

		$\mathbf{X}_{T_w}$-PAM-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entropy
0.0078125	pc.stable	0.2	0.72	0.76	-15697.22	17	12	0.09	2	1.41	43.15
	hc	0.21	0.72	0.76	-15219.23	26	23	0.07	6	1.77	40.91
	mmhc	0.2	0.71	0.76	-15684.39	15	9	0.09	2	1.33	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.015625	pc.stable	0.2	0.72	0.76	-15657	16	10	0.08	2	1.25	43.31
	hc	0.21	0.71	0.76	-15210.4	25	22	0.07	6	1.76	40.99
	mmhc	0.2	0.71	0.76	-15684.39	15	9	0.09	2	1.33	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.03125	pc.stable	0.19	0.71	0.77	-15744.58	15	9	0.09	2	1.2	43.59
	hc	0.21	0.71	0.77	-15185.62	24	22	0.08	6	1.83	40.92
	mmhc	0.2	0.71	0.76	-15684.39	15	9	0.09	2	1.33	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.0625	pc.stable	0.21	0.7	0.74	-15872.25	14	8	0.09	1	1.14	43.98
	hc	0.21	0.71	0.77	-15178.91	23	21	0.08	6	1.83	40.99
	mmhc	0.2	0.71	0.76	-15684.39	15	9	0.09	2	1.33	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.125	pc.stable	0.2	0.71	0.74	-15750.3	14	8	0.09	2	1.14	43.63
	hc	0.21	0.71	0.76	-15123.71	22	20	0.09	6	1.82	41.2
	mmhc	0.21	0.71	0.72	-15684.39	15	9	0.09	2	1.2	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.25	pc.stable	0.17	0.77	0.77	-15827.03	10	6	0.13	2	1.2	43.91
	hc	0.21	0.71	0.76	-15123.71	22	20	0.09	6	1.82	41.2
	mmhc	0.21	0.71	0.72	-15684.39	15	9	0.09	2	1.2	43.41
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.63
0.5	pc.stable	0.2	0.74	0.73	-15753.81	12	7	0.11	2	1.17	43.67
	hc	0.21	0.71	0.75	-15117.65	21	18	0.09	6	1.71	41.39
	mmhc	0.2	0.71	0.74	-15750.3	14	8	0.09	2	1.14	43.63
	aracne	0.2	0.71	0.77	-15446.38	12	10	0.15	5	2	42.71
1	pc.stable	0.2	0.74	0.73	-15753.81	12	7	0.11	2	1.17	43.67
	hc	0.23	0.67	0.71	-15269.58	17	13	0.1	5	1.53	42.11
	mmhc	0.2	0.71	0.74	-15750.3	14	8	0.09	2	1.14	43.63
	aracne	0.21	0.64	0.78	-15446.38	12	10	0.15	5	2.67	42.71
2	pc.stable	0.21	0.69	0.75	-15753.81	12	7	0.11	2	1.33	43.67
	hc	0.22	0.68	0.72	-15301.06	15	12	0.11	5	1.6	42.23
	mmhc	0.19	0.72	0.75	-15781.78	12	7	0.11	2	1.17	43.75
	aracne	0.21	0.64	0.78	-15446.38	12	10	0.15	5	3.33	42.71
4	pc.stable	0.19	0.7	0.89	-16054.47	9	5	0.14	1	1.33	44.54
	hc	0.19	0.71	0.76	-15507.19	12	9	0.14	3	1.67	42.91
	mmhc	0.17	0.76	0.83	-15827.03	10	6	0.13	2	1.6	43.91
	aracne	0.22	0.69	0.78	-15607.26	11	8	0.15	3	2.91	43.23

Table 19

		$\mathbf{X}_{T_w}$-PAM-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entropy
0.001953125	pc.stable	0.28	0.59	0.63	-16359.37	18	11	0.07	1	1.22	45.19
	hc	0.31	0.57	0.67	-19695.72	32	42	0.08	1	2.62	42.08
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.00390625	pc.stable	0.28	0.61	0.64	-16398.84	13	8	0.1	1	1.23	45.41
	hc	0.32	0.54	0.65	-18170.02	31	36	0.08	1	2.32	42.46
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.0078125	pc.stable	0.27	0.62	0.64	-16392.55	13	7	0.09	1	1.08	45.46
	hc	0.28	0.55	0.69	-17711.99	32	32	0.06	1	2	42.65
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.015625	pc.stable	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	hc	0.28	0.56	0.67	-17698.21	30	29	0.07	1	1.93	43.08
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.03125	pc.stable	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	hc	0.27	0.57	0.68	-16200.91	28	22	0.06	1	1.57	43.27
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.0625	pc.stable	0.26	0.6	0.65	-16586.78	7	4	0.19	1	1.14	46.1
	hc	0.26	0.61	0.7	-16130.61	26	19	0.06	1	1.46	43.39
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.125	pc.stable	0.26	0.6	0.65	-16586.78	7	4	0.19	1	1.14	46.1
	hc	0.27	0.59	0.69	-16097.79	23	16	0.06	1	1.39	43.85
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.25	pc.stable	0.26	0.68	0.63	-16730.95	2	1	1	1	1	46.61
	hc	0.27	0.6	0.67	-16041.59	21	14	0.07	1	1.33	44.09
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.5	pc.stable	0.3	0.61	0.58	-16657.73	4	2	0.33	1	1	46.37
	hc	0.27	0.6	0.67	-16045.96	20	13	0.07	1	1.3	44.15
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
1	pc.stable	0.3	0.61	0.58	-16657.73	4	2	0.33	1	1	46.37
	hc	0.26	0.61	0.68	-16141.27	17	10	0.07	1	1.18	44.65
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37

	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
2	pc.stable	0.3	0.61	0.58	-16657.73	4	2	0.33	1	1	46.37
	hc	0.26	0.62	0.69	-16215.01	14	8	0.09	1	1.14	44.92
	mmhc	0.28	0.54	0.62	-16702.3	4	2	0.33	1	1	46.49
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
4	pc.stable	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
	hc	0.3	0.55	0.6	-16713.39	2	1	1	1	1	46.56
	mmhc	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87

Table 20

		$\mathbf{X}_{T_w}$-interval-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.08	0.9	0.83	-11317.26	14	9	0.1	2	1.29	30.97
	hc	0.08	0.88	0.84	-11549.99	25	26	0.09	5	2.08	29.04
	mmhc	0.06	0.92	0.91	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.06	0.93	0.92	-11262.78	10	8	0.18	2	1.6	30.99
0.015625	pc.stable	0.08	0.92	0.81	-11336.13	9	6	0.17	3	1.33	31.28
	hc	0.08	0.89	0.84	-11177.57	24	23	0.08	5	1.92	29.1
	mmhc	0.06	0.92	0.91	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.06	0.93	0.92	-11262.78	10	8	0.18	2	1.6	30.99
0.03125	pc.stable	0.07	0.92	0.81	-11274.38	11	7	0.13	2	1.27	31.07
	hc	0.08	0.89	0.83	-11051.53	24	22	0.08	5	1.83	29.28
	mmhc	0.07	0.91	0.84	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.06	0.93	0.92	-11262.78	10	8	0.18	2	1.6	30.99
0.0625	pc.stable	0.07	0.92	0.84	-11336.13	9	6	0.17	2	1.33	31.28
	hc	0.08	0.89	0.81	-11038	24	20	0.07	4	1.67	29.4
	mmhc	0.07	0.91	0.84	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.06	0.93	0.92	-11262.78	10	8	0.18	2	1.6	30.99
0.125	pc.stable	0.07	0.92	0.81	-11360.17	10	6	0.13	2	1.2	31.35
	hc	0.09	0.87	0.8	-11083.13	22	18	0.08	3	1.64	29.6
	mmhc	0.07	0.91	0.84	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.06	0.93	0.92	-11262.78	10	8	0.18	2	1.6	30.99
0.25	pc.stable	0.06	0.93	0.91	-11331.52	12	8	0.12	2	1.33	31.19
	hc	0.09	0.89	0.8	-11003.85	20	16	0.08	3	1.6	29.82
	mmhc	0.07	0.91	0.84	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.05	0.93	0.94	-11262.78	10	8	0.18	2	1.6	30.99
0.5	pc.stable	0.07	0.92	0.89	-11461.14	12	8	0.12	2	1.33	31.3
	hc	0.08	0.9	0.83	-11015.93	18	13	0.08	3	1.44	30.08
	mmhc	0.08	0.9	0.8	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.08	0.91	0.94	-11250.44	10	7	0.16	3	2.4	31.01
1	pc.stable	0.08	0.91	0.87	-11461.14	12	8	0.12	2	1.5	31.3
	hc	0.07	0.92	0.85	-11053.76	15	11	0.1	3	1.47	30.33
	mmhc	0.07	0.92	0.86	-11336.13	9	6	0.17	2	1.33	31.28

	aracne	0.08	0.9	0.94	-11250.44	10	7	0.16	3	2.8	31.01
2	pc.stable	0.08	0.91	0.87	-11461.14	12	8	0.12	2	1.5	31.3
	hc	0.08	0.91	0.83	-11175.83	13	10	0.13	3	1.54	30.68
	mmhc	0.07	0.92	0.86	-11336.13	9	6	0.17	2	1.33	31.28
	aracne	0.08	0.9	0.94	-11250.44	10	7	0.16	3	2.8	31.01
4	pc.stable	0.07	0.92	0.92	-11515.76	12	8	0.12	1	1.5	31.38
	hc	0.1	0.91	0.89	-11640.27	4	2	0.33	1	2	32.26
	mmhc	0.1	0.91	0.89	-11640.27	4	2	0.33	1	2	32.26
	aracne	0.1	0.91	0.89	-11640.27	4	2	0.33	1	2	32.26

Table 21

		$\mathbf{X}_{T_w}$-interval-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.14	0.79	0.61	-11735.59	14	10	0.11	1	1.43	32.1
	hc	0.12	0.84	0.63	-12154.65	27	24	0.07	1	1.78	30.96
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.015625	pc.stable	0.11	0.87	0.62	-11770.78	8	5	0.18	1	1.25	32.49
	hc	0.11	0.86	0.72	-11682.89	24	18	0.07	1	1.5	31.17
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.03125	pc.stable	0.11	0.87	0.74	-11856.78	9	6	0.17	1	1.33	32.48
	hc	0.12	0.84	0.75	-11728.38	23	18	0.07	1	1.57	31.19
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.0625	pc.stable	0.12	0.85	0.62	-11751.75	7	4	0.19	1	1.14	32.5
	hc	0.13	0.83	0.72	-11725.43	20	15	0.08	1	1.5	31.32
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.125	pc.stable	0.14	0.84	0.62	-11796.39	5	3	0.3	1	1.2	32.66
	hc	0.11	0.85	0.72	-11735.78	19	14	0.08	1	1.47	31.38
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.25	pc.stable	0.14	0.84	0.62	-11796.39	5	3	0.3	1	1.2	32.66
	hc	0.12	0.85	0.63	-11634	15	10	0.1	1	1.33	31.86
	mmhc	0.04	0.96	NaN	-11883.43	4	2	0.33	1	1	32.94
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
0.5	pc.stable	0.14	0.84	0.62	-11796.39	5	3	0.3	1	1.2	32.66
	hc	0.11	0.85	0.67	-11625.52	14	8	0.09	1	1.14	32.02
	mmhc	0.03	0.97	NaN	-11912.63	2	1	1	1	1	33.06
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
1	pc.stable	0.1	0.88	0.61	-11846.08	4	2	0.33	1	1	32.84
	hc	0.11	0.85	0.67	-11625.52	14	8	0.09	1	1.14	32.02

	mmhc	0.03	0.97	NaN	-11912.63	2	1	1	1	1	33.06
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
2	pc.stable	0.1	0.88	0.61	-11846.08	4	2	0.33	1	1	32.84
	hc	0.12	0.84	0.61	-11703.84	8	5	0.18	1	1.25	32.34
	mmhc	0.03	0.97	NaN	-11912.63	2	1	1	1	1	33.06
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
4	pc.stable	0.03	0.97	NaN	-11912.63	2	1	1	1	1	33.06
	hc	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
	mmhc	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21
	aracne	NA	NA	NaN	-11958.14	0	0	NaN	0	NaN	33.21

Table 22

		$\mathbf{X}_{T_w}$-quantile-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entropy
0.0078125	pc.stable	0.26	0.65	0.75	-17751.78	15	11	0.1	3	1.47	48.86
	hc	0.26	0.62	0.75	-17324.42	23	21	0.08	3	1.83	46.92
	mmhc	0.23	0.68	0.75	-17702.66	14	10	0.11	3	1.43	49
	aracne	0.25	0.65	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.015625	pc.stable	0.26	0.65	0.74	-17554.95	14	10	0.11	3	1.43	48.56
	hc	0.26	0.61	0.75	-17308.39	24	22	0.08	3	1.83	46.92
	mmhc	0.23	0.69	0.75	-17702.66	14	10	0.11	3	1.43	49
	aracne	0.25	0.65	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.03125	pc.stable	0.25	0.67	0.74	-17459.99	14	10	0.11	3	1.43	48.37
	hc	0.26	0.61	0.74	-17287.96	23	20	0.08	3	1.74	47.11
	mmhc	0.23	0.69	0.75	-17702.66	14	10	0.11	3	1.43	49
	aracne	0.25	0.64	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.0625	pc.stable	0.24	0.68	0.75	-17579.02	13	9	0.12	3	1.38	48.74
	hc	0.26	0.6	0.75	-17308.39	24	22	0.08	3	1.83	46.92
	mmhc	0.23	0.69	0.75	-17702.66	14	10	0.11	3	1.43	49
	aracne	0.25	0.64	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.125	pc.stable	0.23	0.7	0.75	-17614.18	14	9	0.1	2	1.29	48.84
	hc	0.25	0.6	0.75	-17260.21	23	20	0.08	3	1.74	47.17
	mmhc	0.23	0.68	0.75	-17702.66	14	10	0.11	3	1.43	49
	aracne	0.25	0.65	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.25	pc.stable	0.22	0.71	0.74	-17609.71	15	9	0.09	2	1.2	48.83
	hc	0.25	0.59	0.74	-17223.35	22	18	0.08	3	1.64	47.37
	mmhc	0.23	0.69	0.75	-17686.86	13	9	0.12	2	1.38	49.04
	aracne	0.25	0.65	0.78	-17446.17	12	10	0.15	3	1.67	48.33
0.5	pc.stable	0.24	0.61	0.79	-17839.81	11	7	0.13	2	1.64	49.46
	hc	0.25	0.58	0.75	-17244.17	20	16	0.08	3	1.6	47.57
	mmhc	0.21	0.71	0.77	-17800.42	12	8	0.12	2	1.33	49.39
	aracne	0.27	0.57	0.78	-17446.17	12	10	0.15	5	2.17	48.33
1	pc.stable	0.24	0.61	0.79	-17839.81	11	7	0.13	2	1.64	49.46
	hc	0.25	0.64	0.76	-17392.6	15	11	0.1	3	1.47	48.15

	mmhc	0.21	0.72	0.77	-17800.42	12	8	0.12	2	1.33	49.39
	aracne	0.26	0.57	0.79	-17537.17	12	9	0.14	3	2.67	48.62
2	pc.stable	0.24	0.59	0.82	-17947.72	11	7	0.13	2	1.64	49.86
	hc	0.25	0.64	0.77	-17485.71	14	10	0.11	2	1.43	48.44
	mmhc	0.2	0.73	0.79	-17822.68	11	7	0.13	2	1.27	49.49
	aracne	0.27	0.54	0.79	-17537.17	12	9	0.14	3	3	48.62
4	pc.stable	0.24	0.59	0.82	-17947.72	11	7	0.13	2	1.64	49.86
	hc	0.22	0.74	0.8	-17704.23	10	7	0.16	2	1.4	49.16
	mmhc	0.21	0.65	0.86	-17953.36	8	5	0.18	3	2.25	49.92
	aracne	0.26	0.58	0.8	-17704.23	10	7	0.16	3	2.8	49.16

Table 23

		$\mathbf{X}_{T_w}$-quantile-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entropy
0.0078125	pc.stable	0.37	0.41	0.61	-18591.97	16	9	0.07	1	1.12	51.39
	hc	0.37	0.44	0.59	-18518.82	25	27	0.09	1	2.16	48.19
	mmhc	0.35	0.41	0.57	-18736.66	6	3	0.2	1	1	52.19
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.015625	pc.stable	0.39	0.44	0.59	-18529.26	13	7	0.09	1	1.08	51.42
	hc	0.37	0.42	0.58	-18573.08	25	25	0.08	1	2	48.6
	mmhc	0.35	0.41	0.57	-18736.66	6	3	0.2	1	1	52.19
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.03125	pc.stable	0.33	0.5	0.57	-18613.78	8	4	0.14	1	1	51.81
	hc	0.34	0.46	0.59	-18080.55	24	22	0.08	1	1.83	48.83
	mmhc	0.35	0.41	0.57	-18736.66	6	3	0.2	1	1	52.19
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.0625	pc.stable	0.32	0.55	0.6	-18551.83	8	4	0.14	1	1	51.63
	hc	0.34	0.46	0.59	-18080.55	24	22	0.08	1	1.83	48.83
	mmhc	0.35	0.41	0.57	-18736.66	6	3	0.2	1	1	52.19
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.125	pc.stable	0.32	0.55	0.6	-18551.83	8	4	0.14	1	1	51.63
	hc	0.34	0.48	0.59	-18039.53	22	20	0.09	1	1.82	48.92
	mmhc	0.35	0.41	0.57	-18736.66	6	3	0.2	1	1	52.19
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.25	pc.stable	0.32	0.55	0.6	-18551.83	8	4	0.14	1	1	51.63
	hc	0.33	0.48	0.59	-18026.87	20	18	0.09	1	1.8	49.07
	mmhc	0.35	0.49	0.57	-18730.47	4	2	0.33	1	1	52.2
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
0.5	pc.stable	0.32	0.57	0.61	-18603.28	6	3	0.2	1	1	51.81
	hc	0.33	0.45	0.59	-18025.51	20	17	0.09	1	1.7	49.17
	mmhc	0.35	0.49	0.57	-18730.47	4	2	0.33	1	1	52.2
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
1	pc.stable	0.32	0.57	0.61	-18603.28	6	3	0.2	1	1	51.81

	hc	0.35	0.44	0.56	-18182.78	16	10	0.08	1	1.25	50.4
	mmhc	0.35	0.49	0.57	-18730.47	4	2	0.33	1	1	52.2
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
2	pc.stable	0.33	0.57	0.61	-18686.98	4	2	0.33	1	1	52.08
	hc	0.35	0.44	0.57	-18222.03	15	9	0.09	1	1.2	50.54
	mmhc	0.35	0.49	0.57	-18730.47	4	2	0.33	1	1	52.2
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73
4	pc.stable	0.33	0.57	0.61	-18686.98	4	2	0.33	1	1	52.08
	hc	0.38	0.48	0.57	-18781.92	2	1	1	1	1	52.38
	mmhc	0.38	0.48	0.57	-18781.92	2	1	1	1	1	52.38
	aracne	NA	NA	NaN	-18895.48	0	0	NaN	0	NaN	52.73

Table 24

		$\mathbf{X}_{T_w}$-hartemink-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.22	0.68	0.81	-15968.04	17	13	0.1	2	1.53	43.86
	hc	0.22	0.68	0.78	-15510.26	21	21	0.1	5	2	41.91
	mmhc	0.21	0.72	0.81	-15848.18	17	13	0.1	2	1.53	43.6
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.015625	pc.stable	0.19	0.73	0.8	-15891.87	15	11	0.1	2	1.47	43.85
	hc	0.2	0.69	0.8	-15514.21	19	18	0.11	4	1.89	42.37
	mmhc	0.21	0.71	0.81	-15848.18	17	13	0.1	2	1.53	43.6
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.03125	pc.stable	0.21	0.69	0.75	-15914.43	12	8	0.12	2	1.33	44.06
	hc	0.22	0.69	0.78	-15416.01	19	19	0.11	5	2	42.01
	mmhc	0.21	0.71	0.81	-15848.18	17	13	0.1	2	1.53	43.6
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.0625	pc.stable	0.21	0.69	0.74	-16059.63	12	7	0.11	1	1.17	44.55
	hc	0.21	0.69	0.75	-15434.79	17	16	0.12	5	1.88	42.25
	mmhc	0.22	0.67	0.75	-15898.86	17	12	0.09	2	1.41	43.78
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.125	pc.stable	0.17	0.74	0.79	-16100.11	10	6	0.13	2	1.2	44.69
	hc	0.21	0.71	0.76	-15460.59	16	15	0.12	5	1.88	42.34
	mmhc	0.19	0.73	0.8	-15898.13	16	10	0.08	2	1.25	43.92
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.25	pc.stable	0.17	0.74	0.79	-16100.11	10	6	0.13	2	1.2	44.69
	hc	0.21	0.71	0.75	-15420.75	15	13	0.12	5	1.73	42.59
	mmhc	0.19	0.73	0.8	-15898.13	16	10	0.08	2	1.25	43.92
	aracne	0.21	0.71	0.75	-15672.1	12	9	0.14	3	1.5	43.39
0.5	pc.stable	0.2	0.71	0.79	-15950.08	11	7	0.13	2	1.45	44.24
	hc	0.21	0.71	0.75	-15425.58	14	12	0.13	5	1.71	42.63
	mmhc	0.18	0.75	0.79	-16041.62	12	7	0.11	2	1.17	44.52
	aracne	0.22	0.66	0.75	-15672.1	12	9	0.14	3	1.67	43.39
	pc.stable	0.21	0.69	0.76	-16104.94	9	5	0.14	2	1.33	44.73

1

	hc	0.22	0.7	0.74	-15522.33	13	10	0.13	5	1.54	42.93
	mmhc	0.2	0.72	0.75	-16064.45	11	6	0.11	2	1.09	44.58
	aracne	0.24	0.63	0.72	-15672.1	12	9	0.14	3	3	43.39
2	pc.stable	0.19	0.74	0.8	-16082.11	10	6	0.13	1	1.2	44.66
	hc	0.22	0.7	0.74	-15522.33	13	10	0.13	5	1.54	42.93
	mmhc	0.18	0.74	0.79	-16104.94	9	5	0.14	2	1.11	44.73
	aracne	0.24	0.63	0.72	-15672.1	12	9	0.14	3	3	43.39
4	pc.stable	0.19	0.74	0.81	-16077.29	11	7	0.13	1	1.27	44.63
	hc	0.22	0.7	0.74	-15614.63	12	9	0.14	5	1.5	43.22
	mmhc	0.18	0.74	0.79	-16104.94	9	5	0.14	2	1.33	44.73
	aracne	0.26	0.63	0.73	-15862.87	11	7	0.13	3	2.55	43.99

Table 25

		$\mathbf{X}_{T_w}$-hartemink-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.27	0.6	0.7	-16336.71	18	13	0.08	1	1.44	44.79
	hc	0.31	0.56	0.64	-16085.61	24	22	0.08	1	1.83	43.14
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.015625	pc.stable	0.25	0.6	0.71	-16353.23	14	10	0.11	1	1.43	45.31
	hc	0.32	0.54	0.61	-16122.15	24	20	0.07	1	1.67	43.56
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.03125	pc.stable	0.24	0.65	0.72	-16511.28	11	7	0.13	1	1.27	45.88
	hc	0.31	0.57	0.63	-16137.42	22	19	0.08	1	1.73	43.57
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.0625	pc.stable	0.25	0.64	0.68	-16504.57	11	6	0.11	1	1.09	45.93
	hc	0.3	0.58	0.65	-16091.8	20	15	0.08	1	1.5	44.01
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.125	pc.stable	0.23	0.68	0.76	-16455.06	8	4	0.14	1	1	45.73
	hc	0.3	0.57	0.68	-16057.51	20	16	0.08	1	1.6	43.82
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.25	pc.stable	0.23	0.68	0.76	-16455.06	8	4	0.14	1	1	45.73
	hc	0.29	0.6	0.67	-16008.26	20	13	0.07	1	1.3	44.05
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.5	pc.stable	0.23	0.68	0.76	-16455.06	8	4	0.14	1	1	45.73
	hc	0.29	0.6	0.66	-16011.2	19	12	0.07	1	1.26	44.16
	mmhc	0.27	0.59	0.66	-16619.03	10	5	0.11	1	1	46.16
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2

1	pc.stable	0.23	0.68	0.76	-16455.06	8	4	0.14	1	1	45.73
	hc	0.29	0.6	0.66	-16072.37	16	9	0.07	1	1.12	44.49
	mmhc	0.28	0.57	0.66	-16623.79	8	4	0.14	1	1	46.21
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
2	pc.stable	0.19	0.76	0.83	-16538.89	6	3	0.2	1	1	46
	hc	0.29	0.6	0.66	-16072.37	16	9	0.07	1	1.12	44.49
	mmhc	0.22	0.73	0.8	-16688.92	4	2	0.33	1	1	46.46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
4	pc.stable	0.19	0.76	0.83	-16538.89	6	3	0.2	1	1	46
	hc	0.23	0.7	0.79	-16415.73	8	4	0.14	1	1	45.62
	mmhc	0.22	0.73	0.8	-16688.92	4	2	0.33	1	1	46.46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2

Table 26

		$\mathbf{X}_{T_d}$-PAM-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.16	0.79	0.82	-20840.62	22	17	0.07	4	1.55	56.4
	hc	0.16	0.8	0.81	-20969.15	20	18	0.09	7	1.8	56.48
	mmhc	0.15	0.81	0.82	-20913.39	20	15	0.08	4	1.5	56.6
	aracne	0.19	0.74	0.77	-21482.03	10	8	0.18	6	1.6	58.45
0.015625	pc.stable	0.15	0.81	0.83	-20862.75	21	16	0.08	4	1.52	56.48
	hc	0.16	0.8	0.81	-20969.15	20	18	0.09	7	1.8	56.48
	mmhc	0.15	0.81	0.8	-20920.76	20	15	0.08	4	1.5	56.63
	aracne	0.19	0.74	0.77	-21482.03	10	8	0.18	6	1.6	58.45
0.03125	pc.stable	0.16	0.77	0.82	-20921.96	21	15	0.07	3	1.43	56.77
	hc	0.16	0.8	0.81	-20969.15	20	18	0.09	7	1.8	56.48
	mmhc	0.15	0.81	0.8	-20920.76	20	15	0.08	4	1.5	56.63
	aracne	0.19	0.74	0.77	-21482.03	10	8	0.18	6	1.6	58.45
0.0625	pc.stable	0.16	0.79	0.8	-20959.95	20	13	0.07	3	1.3	56.87
	hc	0.16	0.8	0.81	-20969.15	20	18	0.09	7	1.8	56.48
	mmhc	0.16	0.8	0.8	-20998.98	20	15	0.08	4	1.5	56.9
	aracne	0.19	0.74	0.77	-21482.03	10	8	0.18	6	1.6	58.45
0.125	pc.stable	0.15	0.81	0.81	-21092.93	18	11	0.07	2	1.22	57.27
	hc	0.16	0.8	0.81	-20892.57	20	17	0.09	7	1.7	56.51
	mmhc	0.16	0.8	0.8	-20998.98	20	15	0.08	4	1.5	56.9
	aracne	0.19	0.74	0.77	-21482.03	10	8	0.18	6	1.6	58.45
0.25	pc.stable	0.14	0.82	0.82	-21174.64	17	10	0.07	1	1.18	57.52
	hc	0.16	0.8	0.81	-20879.92	20	16	0.08	7	1.6	56.49
	mmhc	0.16	0.8	0.79	-20986.34	20	14	0.07	4	1.4	56.88
	aracne	0.2	0.72	0.77	-21482.03	10	8	0.18	6	1.8	58.45
0.5	pc.stable	0.14	0.82	0.82	-21174.64	17	10	0.07	1	1.18	57.52
	hc	0.14	0.82	0.81	-20949.86	18	12	0.08	3	1.33	56.84
	mmhc	0.14	0.82	0.82	-21052.06	17	11	0.08	2	1.29	57.15

	aracne	0.2	0.73	0.77	-21482.03	10	8	0.18	6	3.2	58.45
1	pc.stable	0.14	0.82	0.82	-21174.64	17	10	0.07	2	1.18	57.52
	hc	0.14	0.81	0.81	-20959.77	17	11	0.08	3	1.29	56.9
	mmhc	0.14	0.81	0.82	-21061.97	16	10	0.08	2	1.25	57.21
	aracne	0.2	0.73	0.77	-21482.03	10	8	0.18	6	3.2	58.45
2	pc.stable	0.14	0.83	0.83	-21221.97	16	9	0.07	1	1.12	57.69
	hc	0.14	0.82	0.81	-21117.32	14	9	0.1	3	1.29	57.42
	mmhc	0.14	0.82	0.82	-21219.52	13	8	0.1	2	1.23	57.73
	aracne	0.21	0.76	0.77	-21510.2	10	7	0.16	3	2.8	58.56
4	pc.stable	0.14	0.83	0.83	-21221.97	16	9	0.07	1	1.12	57.69
	hc	0.13	0.82	0.83	-21164.65	13	8	0.1	2	1.23	57.58
	mmhc	0.13	0.83	0.84	-21266.85	12	7	0.11	2	1.17	57.9
	aracne	0.19	0.75	0.86	-21691.93	6	4	0.27	3	2.67	59.16

Table 27

		$\mathbf{X}_{T_d}$-PAM-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entropy
0.0078125	pc.stable	0.19	0.77	0.76	-21443.81	22	17	0.07	1	1.55	57.66
	hc	0.18	0.76	0.73	-21445.97	23	16	0.06	1	1.39	57.54
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.015625	pc.stable	0.18	0.76	0.77	-21433.35	21	14	0.07	1	1.33	57.99
	hc	0.17	0.76	0.74	-21359.18	22	15	0.06	1	1.36	57.61
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.03125	pc.stable	0.18	0.77	0.79	-21458.1	20	13	0.07	1	1.3	58.09
	hc	0.18	0.75	0.75	-21281.86	22	14	0.06	1	1.27	57.7
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.0625	pc.stable	0.18	0.76	0.76	-21325.28	21	12	0.06	1	1.14	57.94
	hc	0.18	0.75	0.75	-21281.86	22	14	0.06	1	1.27	57.7
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.125	pc.stable	0.18	0.76	0.75	-21286.21	20	12	0.06	1	1.2	57.83
	hc	0.18	0.75	0.75	-21281.86	22	14	0.06	1	1.27	57.7
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.25	pc.stable	0.17	0.78	0.77	-21376.1	17	9	0.07	1	1.06	58.1
	hc	0.18	0.75	0.75	-21281.86	22	14	0.06	1	1.27	57.7
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
0.5	pc.stable	0.16	0.8	0.77	-21572.69	13	7	0.09	1	1.08	58.71
	hc	0.18	0.75	0.75	-21269.06	21	13	0.06	1	1.24	57.75
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55

	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
1	pc.stable	0.16	0.79	0.77	-21572.69	13	7	0.09	1	1.08	58.71
	hc	0.17	0.78	0.74	-21286.87	19	11	0.06	1	1.16	57.79
	mmhc	0.16	0.8	0.76	-21525.13	15	8	0.08	1	1.07	58.55
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
2	pc.stable	0.16	0.77	0.79	-21564.23	14	8	0.09	1	1.14	58.73
	hc	0.17	0.78	0.74	-21377.24	17	9	0.07	1	1.06	58.13
	mmhc	0.17	0.79	0.74	-21581.63	14	7	0.08	1	1	58.76
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09
4	pc.stable	0.16	0.77	0.79	-21564.23	14	8	0.09	1	1.14	58.73
	hc	0.17	0.78	0.75	-21506.51	13	7	0.09	1	1.08	58.55
	mmhc	0.17	0.79	0.76	-21710.9	10	5	0.11	1	1	59.18
	aracne	NA	NA	NaN	-22346.41	0	0	NaN	0	NaN	61.09

Table 28

		$\mathbf{X}_{T_d}$-interval-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.04	0.96	0.86	-11105.52	16	10	0.08	2	1.25	29.78
	hc	0.04	0.95	0.84	-11002.48	21	16	0.08	4	1.52	29.21
	mmhc	0.04	0.95	0.87	-11153.07	16	11	0.09	3	1.38	29.78
	aracne	0.03	0.96	0.89	-11162.91	11	8	0.15	2	1.45	29.97
0.015625	pc.stable	0.04	0.94	0.86	-11108.37	12	8	0.12	2	1.33	29.85
	hc	0.05	0.94	0.84	-11028.02	20	15	0.08	4	1.5	29.31
	mmhc	0.04	0.95	0.86	-11155.55	16	11	0.09	2	1.38	29.79
	aracne	0.03	0.96	0.89	-11162.91	11	8	0.15	2	1.45	29.97
0.03125	pc.stable	0.05	0.93	0.86	-11144.87	11	7	0.13	2	1.27	29.98
	hc	0.05	0.94	0.84	-11028.02	20	15	0.08	4	1.5	29.31
	mmhc	0.04	0.95	0.86	-11125.19	15	10	0.1	2	1.33	29.79
	aracne	0.03	0.96	0.89	-11162.91	11	8	0.15	2	1.45	29.97
0.0625	pc.stable	0.05	0.93	0.86	-11144.87	11	7	0.13	2	1.27	29.98
	hc	0.05	0.94	0.84	-11028.02	20	15	0.08	4	1.5	29.31
	mmhc	0.04	0.95	0.86	-11125.19	15	10	0.1	2	1.33	29.79
	aracne	0.03	0.96	0.89	-11162.91	11	8	0.15	2	1.45	29.97
0.125	pc.stable	0.05	0.93	0.86	-11192.74	10	6	0.13	2	1.2	30.15
	hc	0.05	0.94	0.84	-11028.55	18	14	0.09	4	1.56	29.35
	mmhc	0.04	0.95	0.86	-11125.19	15	10	0.1	2	1.33	29.79
	aracne	0.03	0.96	0.89	-11162.91	11	8	0.15	2	1.64	29.97
0.25	pc.stable	0.05	0.93	0.86	-11192.74	10	6	0.13	2	1.2	30.15
	hc	0.04	0.95	0.8	-10992.18	15	11	0.1	3	1.47	29.44
	mmhc	0.05	0.93	0.86	-11151.36	9	6	0.17	2	1.33	30.04
	aracne	0.05	0.95	0.63	-11137.96	11	8	0.15	3	2.73	29.95
0.5	pc.stable	0.04	0.94	0.86	-11196.28	12	7	0.11	2	1.17	30.12
	hc	0.04	0.95	0.79	-11092.1	11	7	0.13	2	1.27	29.86

	mmhc	0.05	0.94	0.87	-11185.87	7	4	0.19	2	1.14	30.21
	aracne	0.05	0.95	0.63	-11137.96	11	8	0.15	3	2.91	29.95
1	pc.stable	0.04	0.94	0.86	-11196.28	12	7	0.11	2	1.17	30.12
	hc	0.04	0.95	0.79	-11092.1	11	7	0.13	2	1.27	29.86
	mmhc	0.05	0.94	0.87	-11185.87	7	4	0.19	2	1.14	30.21
	aracne	0.04	0.96	0.7	-11145.59	10	7	0.16	3	2.8	30
2	pc.stable	0.04	0.94	0.86	-11202.77	10	6	0.13	2	1.2	30.17
	hc	0.06	0.92	0.79	-11187.96	6	4	0.27	2	1.33	30.22
	mmhc	0.05	0.94	0.87	-11185.87	7	4	0.19	2	1.14	30.21
	aracne	0.07	0.94	0.66	-11302.89	4	2	0.33	1	2	30.6
4	pc.stable	0.04	0.94	0.86	-11202.77	10	6	0.13	2	1.2	30.17
	hc	0.05	0.97	0.64	-11350.76	2	1	1	1	2	30.76
	mmhc	0.05	0.97	0.64	-11350.76	2	1	1	1	2	30.76
	aracne	0.05	0.97	0.64	-11350.76	2	1	1	1	2	30.76

Table 29

		$\mathbf{X}_{T_d}$-interval-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.04	0.95	0.73	-11320.64	17	10	0.07	1	1.18	30.15
	hc	0.05	0.94	0.75	-11305.4	26	19	0.06	1	1.46	29.6
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.015625	pc.stable	0.05	0.94	0.77	-11254.58	10	5	0.11	1	1	30.36
	hc	0.04	0.95	0.74	-11191.51	25	17	0.06	1	1.36	29.63
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.03125	pc.stable	0.05	0.94	0.77	-11254.58	10	5	0.11	1	1	30.36
	hc	0.04	0.95	0.77	-11199.99	23	16	0.06	1	1.39	29.69
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.0625	pc.stable	0.06	0.92	0.77	-11296.79	8	4	0.14	1	1	30.51
	hc	0.04	0.95	0.77	-11178.61	23	15	0.06	1	1.3	29.71
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.125	pc.stable	0.06	0.93	0.77	-11296.79	8	4	0.14	1	1	30.51
	hc	0.04	0.95	0.77	-11178.61	23	15	0.06	1	1.3	29.71
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.25	pc.stable	0.06	0.93	0.77	-11296.79	8	4	0.14	1	1	30.51
	hc	0.04	0.94	0.72	-11164.99	16	9	0.07	1	1.12	29.94
	mmhc	0.04	0.94	0.77	-11273.17	12	6	0.09	1	1	30.38
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
0.5	pc.stable	0.05	0.94	0.77	-11306.81	8	4	0.14	1	1	30.54
	hc	0.05	0.93	0.72	-11210.43	11	6	0.11	1	1.09	30.21

	mmhc	0.05	0.92	0.77	-11269.63	10	5	0.11	1	1	30.41
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
1	pc.stable	0.05	0.94	0.77	-11306.81	8	4	0.14	1	1	30.54
	hc	0.05	0.93	0.72	-11210.43	11	6	0.11	1	1.09	30.21
	mmhc	0.06	0.92	0.77	-11276.12	8	4	0.14	1	1	30.46
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
2	pc.stable	0.05	0.94	0.77	-11306.81	8	4	0.14	1	1	30.54
	hc	0.08	0.9	0.69	-11345.62	4	2	0.33	1	1	30.72
	mmhc	0.05	0.94	0.76	-11343.54	4	2	0.33	1	1	30.71
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
4	pc.stable	0.05	0.94	0.77	-11306.81	8	4	0.14	1	1	30.54
	hc	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
	mmhc	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1
	aracne	NA	NA	NaN	-11460.56	0	0	NaN	0	NaN	31.1

Table 30

		$\mathbf{X}_{T_d}$-quantile-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.21	0.71	0.83	-23418.94	24	19	0.07	4	1.58	63.53
	hc	0.22	0.69	0.78	-23312.54	27	22	0.06	3	1.63	62.75
	mmhc	0.22	0.7	0.79	-23243.3	23	18	0.07	4	1.57	63.02
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
0.015625	pc.stable	0.22	0.71	0.81	-23326.58	24	17	0.06	4	1.42	63.37
	hc	0.22	0.68	0.79	-23304.05	26	22	0.07	3	1.69	62.75
	mmhc	0.22	0.7	0.79	-23243.3	23	18	0.07	4	1.57	63.02
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
0.03125	pc.stable	0.22	0.71	0.8	-23340.48	23	15	0.06	4	1.3	63.45
	hc	0.22	0.68	0.78	-23297.11	26	22	0.07	6	1.69	62.75
	mmhc	0.22	0.7	0.79	-23243.3	23	18	0.07	4	1.57	63.02
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
0.0625	pc.stable	0.22	0.7	0.78	-23392.9	21	13	0.06	4	1.24	63.62
	hc	0.21	0.71	0.8	-23305.08	24	22	0.08	6	1.83	62.72
	mmhc	0.22	0.7	0.79	-23243.3	23	18	0.07	4	1.57	63.02
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
0.125	pc.stable	0.21	0.72	0.79	-23507.4	18	11	0.07	2	1.22	64.01
	hc	0.22	0.69	0.79	-23235.35	24	21	0.08	6	1.75	62.87
	mmhc	0.22	0.7	0.79	-23228.17	22	17	0.07	4	1.55	62.98
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
0.25	pc.stable	0.21	0.72	0.79	-23491.42	18	12	0.08	2	1.33	63.93
	hc	0.22	0.71	0.79	-23266.8	23	20	0.08	4	1.74	62.98
	mmhc	0.22	0.7	0.79	-23204.37	22	16	0.07	4	1.45	63
	aracne	0.2	0.74	0.81	-24020.27	8	6	0.21	2	1.5	65.54
	pc.stable	0.21	0.72	0.8	-23450.23	19	13	0.08	2	1.37	63.79

0.5

	hc	0.22	0.71	0.78	-23234.36	20	15	0.08	3	1.5	63.05
	mmhc	0.23	0.69	0.78	-23353.13	19	13	0.08	3	1.37	63.46
	aracne	0.23	0.6	0.81	-24020.27	8	6	0.21	3	3	65.54
1	pc.stable	0.21	0.72	0.81	-23439	20	14	0.07	2	1.4	63.75
	hc	0.21	0.72	0.78	-23388.05	16	11	0.09	3	1.38	63.63
	mmhc	0.21	0.72	0.78	-23481.89	15	10	0.1	3	1.33	63.92
	aracne	0.23	0.6	0.81	-24020.27	8	6	0.21	3	3	65.54
2	pc.stable	0.21	0.71	0.81	-23541.56	20	13	0.07	2	1.3	64.07
	hc	0.21	0.72	0.78	-23388.05	16	11	0.09	3	1.38	63.63
	mmhc	0.21	0.72	0.78	-23481.89	15	10	0.1	3	1.33	63.92
	aracne	0.23	0.6	0.81	-24020.27	8	6	0.21	3	3	65.54
4	pc.stable	0.21	0.68	0.83	-23590.16	19	12	0.07	1	1.37	64.23
	hc	0.2	0.75	0.79	-23621.54	13	8	0.1	2	1.23	64.37
	mmhc	0.2	0.75	0.79	-23621.54	13	8	0.1	2	1.23	64.37
	aracne	0.22	0.61	0.85	-24159.91	6	4	0.27	3	2.67	65.99

Table 31

		$\mathbf{X}_{T_d}$-quantile-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entropy
0.0078125	pc.stable	0.26	0.63	0.74	-23851.02	25	17	0.06	1	1.36	64.51
	hc	0.26	0.62	0.73	-23980.03	30	24	0.06	1	1.6	63.97
	mmhc	0.25	0.64	0.74	-23983.46	18	11	0.07	1	1.22	65.27
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.015625	pc.stable	0.26	0.63	0.74	-23838.06	24	16	0.06	1	1.33	64.48
	hc	0.25	0.63	0.74	-23808.85	30	22	0.05	1	1.47	64.19
	mmhc	0.25	0.64	0.74	-23983.46	18	11	0.07	1	1.22	65.27
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.03125	pc.stable	0.25	0.64	0.75	-23846.61	24	16	0.06	1	1.33	64.53
	hc	0.25	0.63	0.74	-23793.22	29	21	0.05	1	1.45	64.2
	mmhc	0.25	0.64	0.74	-23983.46	18	11	0.07	1	1.22	65.27
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.0625	pc.stable	0.25	0.64	0.75	-23872.15	21	13	0.06	1	1.24	64.94
	hc	0.25	0.63	0.75	-23787.89	29	21	0.05	1	1.45	64.23
	mmhc	0.25	0.64	0.74	-23983.46	18	11	0.07	1	1.22	65.27
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.125	pc.stable	0.25	0.65	0.76	-24049.29	18	11	0.07	1	1.22	65.49
	hc	0.25	0.63	0.75	-23772.05	29	20	0.05	1	1.38	64.29
	mmhc	0.25	0.64	0.74	-23983.46	18	11	0.07	1	1.22	65.27
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.25	pc.stable	0.25	0.63	0.76	-24143.14	16	10	0.08	1	1.25	65.78
	hc	0.25	0.63	0.73	-23770.36	26	16	0.05	1	1.23	64.56
	mmhc	0.25	0.66	0.76	-23979.05	17	10	0.07	1	1.18	65.29
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
0.5	pc.stable	0.25	0.64	0.77	-24089	16	10	0.08	1	1.25	65.62

	hc	0.25	0.63	0.73	-23757.4	25	15	0.05	1	1.2	64.54
	mmhc	0.25	0.66	0.75	-24020.24	16	9	0.07	1	1.12	65.43
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
1	pc.stable	0.24	0.67	0.77	-24101.73	14	9	0.1	1	1.29	65.69
	hc	0.24	0.67	0.73	-23905.74	19	10	0.06	1	1.05	65.09
	mmhc	0.24	0.69	0.74	-24048.95	14	7	0.08	1	1	65.58
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
2	pc.stable	0.24	0.68	0.77	-24101.73	14	9	0.1	1	1.29	65.69
	hc	0.24	0.68	0.74	-23936.65	18	9	0.06	1	1	65.2
	mmhc	0.24	0.69	0.74	-24048.95	14	7	0.08	1	1	65.58
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05
4	pc.stable	0.22	0.71	0.79	-24150.32	12	8	0.12	1	1.33	65.85
	hc	0.23	0.7	0.74	-24214.86	10	5	0.11	1	1	66.1
	mmhc	0.23	0.7	0.74	-24214.86	10	5	0.11	1	1	66.1
	aracne	NA	NA	NaN	-24862.9	0	0	NaN	0	NaN	68.05

Table 32

		$\mathbf{X}_{T_d}$-Hartemink-B_{co}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.19	0.73	0.79	-21111.4	26	19	0.06	4	1.46	56.89
	hc	0.2	0.69	0.79	-21547.85	29	27	0.07	3	1.86	56.41
	mmhc	0.18	0.74	0.8	-21130.7	23	16	0.06	4	1.39	57.13
	aracne	0.15	0.77	0.88	-21948.17	8	5	0.18	2	1.25	59.85
0.015625	pc.stable	0.19	0.73	0.78	-21106.53	24	17	0.06	4	1.42	56.95
	hc	0.2	0.7	0.79	-21540.06	29	25	0.06	3	1.72	56.49
	mmhc	0.18	0.74	0.8	-21130.7	23	16	0.06	4	1.39	57.13
	aracne	0.15	0.77	0.88	-21948.17	8	5	0.18	2	1.25	59.85
0.03125	pc.stable	0.18	0.75	0.81	-21235.96	21	15	0.07	4	1.43	57.38
	hc	0.2	0.7	0.79	-21174.4	27	24	0.07	3	1.78	56.41
	mmhc	0.18	0.74	0.8	-21130.7	23	16	0.06	4	1.39	57.13
	aracne	0.15	0.77	0.88	-21948.17	8	5	0.18	2	1.25	59.85
0.0625	pc.stable	0.18	0.75	0.81	-21235.96	21	15	0.07	4	1.43	57.38
	hc	0.2	0.7	0.79	-21184.85	27	23	0.07	3	1.7	56.44
	mmhc	0.18	0.74	0.8	-21130.7	23	16	0.06	4	1.39	57.13
	aracne	0.15	0.77	0.88	-21948.17	8	5	0.18	2	1.25	59.85
0.125	pc.stable	0.18	0.75	0.81	-21235.96	21	15	0.07	4	1.43	57.38
	hc	0.2	0.71	0.8	-21111.7	26	23	0.07	3	1.77	56.62
	mmhc	0.18	0.74	0.8	-21130.7	23	16	0.06	4	1.39	57.13
	aracne	0.15	0.77	0.88	-21948.17	8	5	0.18	2	1.25	59.85
0.25	pc.stable	0.17	0.76	0.83	-21266.87	20	14	0.07	4	1.4	57.49
	hc	0.2	0.71	0.79	-21156.37	25	21	0.07	3	1.68	56.82
	mmhc	0.19	0.74	0.79	-21186.38	22	15	0.06	4	1.36	57.33
	aracne	0.19	0.7	0.88	-21948.17	8	5	0.18	3	1.5	59.85

0.5	pc.stable	0.17	0.76	0.82	-21258.84	19	12	0.07	4	1.26	57.64
	hc	0.2	0.7	0.78	-21165.77	21	14	0.07	3	1.33	57.4
	mmhc	0.17	0.76	0.81	-21311.91	17	10	0.07	2	1.18	57.94
	aracne	0.22	0.65	0.86	-21948.17	8	5	0.18	3	2.5	59.85
1	pc.stable	0.17	0.76	0.82	-21258.84	19	12	0.07	4	1.26	57.64
	hc	0.19	0.72	0.79	-21273.96	18	11	0.07	3	1.22	57.8
	mmhc	0.17	0.76	0.82	-21394.96	15	9	0.09	2	1.2	58.2
	aracne	0.22	0.65	0.86	-21948.17	8	5	0.18	3	2.5	59.85
2	pc.stable	0.17	0.76	0.83	-21256.46	19	13	0.08	4	1.37	57.52
	hc	0.18	0.74	0.8	-21292.92	16	10	0.08	3	1.25	57.89
	mmhc	0.17	0.76	0.82	-21394.96	15	9	0.09	2	1.2	58.2
	aracne	0.22	0.65	0.86	-21948.17	8	5	0.18	3	2.5	59.85
4	pc.stable	0.17	0.77	0.83	-21352.21	19	12	0.07	2	1.26	57.82
	hc	0.18	0.75	0.8	-21483.29	14	8	0.09	2	1.14	58.48
	mmhc	0.16	0.78	0.81	-21585.33	13	7	0.09	2	1.08	58.79
	aracne	0.23	0.63	0.79	-22138.53	6	3	0.2	1	2	60.44

Table 33

		$\mathbf{X}_{T_d}$-Hartemink-B_{causal}									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_l$$ength$	degree	entropy
0.0078125	pc.stable	0.22	0.7	0.74	-21566.96	26	19	0.06	1	1.46	57.91
	hc	0.23	0.68	0.73	-23098.14	33	30	0.06	1	1.82	57.34
	mmhc	0.21	0.72	0.75	-21558.51	21	11	0.05	1	1.05	58.58
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.015625	pc.stable	0.22	0.7	0.75	-21514.36	24	16	0.06	1	1.33	58
	hc	0.23	0.68	0.74	-22882.94	32	26	0.05	1	1.62	57.52
	mmhc	0.21	0.72	0.75	-21558.51	21	11	0.05	1	1.05	58.58
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.03125	pc.stable	0.22	0.7	0.73	-21435.13	24	14	0.05	1	1.17	58.07
	hc	0.23	0.68	0.73	-21914.72	31	22	0.05	1	1.42	57.57
	mmhc	0.21	0.72	0.75	-21558.51	21	11	0.05	1	1.05	58.58
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.0625	pc.stable	0.22	0.7	0.73	-21435.13	24	14	0.05	1	1.17	58.07
	hc	0.23	0.68	0.73	-21834.87	31	21	0.05	1	1.35	57.7
	mmhc	0.21	0.72	0.75	-21558.51	21	11	0.05	1	1.05	58.58
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.125	pc.stable	0.21	0.71	0.75	-21559.7	23	15	0.06	1	1.3	58.52
	hc	0.23	0.68	0.73	-21478.58	29	19	0.05	1	1.31	57.66
	mmhc	0.21	0.72	0.75	-21558.51	21	11	0.05	1	1.05	58.58
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.25	pc.stable	0.2	0.73	0.78	-21609.57	19	13	0.08	1	1.37	58.72
	hc	0.23	0.67	0.72	-21455.09	26	16	0.05	1	1.23	58.01
	mmhc	0.21	0.72	0.73	-21614.19	20	10	0.05	1	1	58.77

	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
0.5	pc.stable	0.2	0.72	0.78	-21609.57	19	13	0.08	1	1.37	58.72
	hc	0.23	0.69	0.71	-21456.62	24	13	0.05	1	1.08	58.24
	mmhc	0.21	0.72	0.73	-21614.19	20	10	0.05	1	1	58.77
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
1	pc.stable	0.2	0.72	0.77	-21713.99	17	9	0.07	1	1.06	59.07
	hc	0.23	0.68	0.71	-21539.67	22	12	0.05	1	1.09	58.5
	mmhc	0.21	0.71	0.73	-21697.25	18	9	0.06	1	1	59.03
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
2	pc.stable	0.19	0.75	0.79	-21808.31	15	9	0.09	1	1.2	59.25
	hc	0.21	0.71	0.73	-21626.12	17	9	0.07	1	1.06	58.84
	mmhc	0.2	0.73	0.74	-21728.15	16	8	0.07	1	1	59.15
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93
4	pc.stable	0.18	0.78	0.8	-21876.91	13	8	0.1	1	1.23	59.47
	hc	0.21	0.71	0.73	-21768.2	13	7	0.09	1	1.08	59.29
	mmhc	0.2	0.74	0.74	-21870.24	12	6	0.09	1	1	59.6
	aracne	NA	NA	NaN	-22646.54	0	0	NaN	0	NaN	61.93

Table 34

7 Appendix B

		$\mathbf{X}_{T_w}$-PAM-$B_{\text{co}}extra$									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_l ength$	degree	entrop
0.00048828125	pc.stable	0.2	0.71	0.76	-15876.52	17	14	0.1	2	1.65	42.8
	hc	0.22	0.7	0.75	-16500.61	26	27	0.08	6	2.08	40.9
	mmhc	0.23	0.67	0.72	-15503.06	18	13	0.08	6	1.44	42.7
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.6
0.0009765625	pc.stable	0.2	0.72	0.76	-15876.52	17	14	0.1	2	1.65	42.8
	hc	0.22	0.69	0.75	-15475.6	26	28	0.09	6	2.15	40.8
	mmhc	0.23	0.67	0.72	-15503.06	18	13	0.08	6	1.44	42.7
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.6
0.001953125	pc.stable	0.2	0.71	0.76	-15876.52	17	14	0.1	2	1.65	42.8
	hc	0.22	0.7	0.77	-15286.6	28	25	0.07	6	1.79	40.7
	mmhc	0.23	0.67	0.72	-15503.06	18	13	0.08	6	1.44	42.7
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.6
0.00390625	pc.stable	0.2	0.72	0.76	-15876.52	17	14	0.1	2	1.65	42.8
	hc	0.21	0.7	0.77	-15289.46	28	25	0.07	6	1.79	40.7
	mmhc	0.2	0.72	0.77	-15669.98	17	11	0.08	3	1.29	43.2
	aracne	0.2	0.72	0.78	-15442.73	12	11	0.17	3	1.83	42.6

Table 35

		$\mathbf{X}_{T_w}$-hartemink-$B_{\text{co}}extra$

		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entrop
0.00048828125	pc.stable	0.22	0.66	0.81	-16031.91	18	17	0.11	2	1.89	43.49
	hc	0.24	0.66	0.74	-15930.64	26	29	0.09	5	2.23	41.4
	mmhc	0.21	0.69	0.8	-15732.21	18	14	0.09	3	1.56	43.62
	aracne	0.22	0.69	0.74	-15672.1	12	9	0.14	3	1.83	43.39
0.0009765625	pc.stable	0.22	0.67	0.81	-16031.91	18	17	0.11	2	1.89	43.49
	hc	0.24	0.63	0.72	-15965.66	26	28	0.09	4	2.15	41.78
	mmhc	0.21	0.69	0.8	-15732.21	18	14	0.09	3	1.56	43.62
	aracne	0.22	0.69	0.74	-15672.1	12	9	0.14	3	1.83	43.39
0.001953125	pc.stable	0.22	0.66	0.81	-16031.91	18	17	0.11	2	1.89	43.49
	hc	0.23	0.65	0.75	-15785.12	25	24	0.08	5	1.92	41.97
	mmhc	0.21	0.69	0.8	-15732.21	18	14	0.09	3	1.56	43.62
	aracne	0.22	0.69	0.74	-15672.1	12	9	0.14	3	1.83	43.39
0.00390625	pc.stable	0.21	0.69	0.8	-16259.34	18	15	0.1	2	1.67	43.7
	hc	0.22	0.65	0.75	-15544.93	24	23	0.08	4	1.92	42.1
	mmhc	0.21	0.69	0.81	-15848.18	17	13	0.1	2	1.53	43.91
	aracne	0.22	0.69	0.74	-15672.1	12	9	0.14	3	1.83	43.39

Table 36

		$\mathbf{X}_{T_w}$-PAM-B_{causal} extra									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entrop
0.00048828125	pc.stable	0.28	0.59	0.63	-16359.37	18	11	0.07	1	1.22	45.19
	hc	0.35	0.53	0.61	-24569.93	34	51	0.09	1	3	41.87
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.0009765625	pc.stable	0.28	0.59	0.63	-16359.37	18	11	0.07	1	1.22	45.19
	hc	0.33	0.53	0.67	-21274.17	34	47	0.08	1	2.76	41.96
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.001953125	pc.stable	0.28	0.59	0.63	-16359.37	18	11	0.07	1	1.22	45.19
	hc	0.31	0.57	0.67	-19695.72	32	42	0.08	1	2.62	42.08
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87
0.00390625	pc.stable	0.28	0.61	0.64	-16398.84	13	8	0.1	1	1.23	45.41
	hc	0.32	0.54	0.65	-18170.02	31	36	0.08	1	2.32	42.46
	mmhc	0.28	0.57	0.64	-16670.82	6	3	0.2	1	1	46.37
	aracne	NA	NA	NaN	-16813.46	0	0	NaN	0	NaN	46.87

Table 37

		$\mathbf{X}_{T_w}$-hartemink-B_{causal} extra									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entrop
0.00048828125	pc.stable	0.26	0.62	0.72	-16398.27	20	16	0.08	1	1.6	44.54

	hc	0.34	0.49	0.59	-16289.85	30	30	0.07	1	2	42.7
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.0009765625	pc.stable	0.26	0.63	0.72	-16398.27	20	16	0.08	1	1.6	44.5
	hc	0.32	0.53	0.61	-16175.74	29	27	0.07	1	1.86	42.7
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.001953125	pc.stable	0.26	0.62	0.72	-16398.27	20	16	0.08	1	1.6	44.5
	hc	0.31	0.55	0.63	-16037.12	26	24	0.07	1	1.85	42.9
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2
0.00390625	pc.stable	0.27	0.62	0.7	-16396.31	18	14	0.09	1	1.56	44.6
	hc	0.31	0.56	0.64	-16059.54	24	21	0.08	1	1.75	43.2
	mmhc	0.28	0.55	0.66	-16618.47	12	7	0.11	1	1.17	46
	aracne	NA	NA	NaN	-16930.02	0	0	NaN	0	NaN	47.2

Table 38

		$\mathbf{X}_{T_w}$- PAM-B_{co}. 4 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.27	0.54	0.75	-20600.75	15	9	0.09	2	1.2	56.58
	hc	0.31	0.54	0.71	-20733.52	19	19	0.11	3	2	53.72
	mmhc	0.23	0.66	0.81	-20673.69	9	6	0.17	2	1.33	57
	aracne	0.26	0.58	0.77	-20355.23	11	9	0.16	2	1.64	55.69
0.015625	pc.stable	0.27	0.6	0.73	-20602.2	13	8	0.1	2	1.23	56.65
	hc	0.31	0.54	0.71	-20733.52	19	19	0.11	3	2	53.72
	mmhc	0.23	0.65	0.81	-20673.69	9	6	0.17	2	1.33	57
	aracne	0.26	0.59	0.77	-20355.23	11	9	0.16	2	1.64	55.69
0.03125	pc.stable	0.28	0.59	0.73	-20665.48	13	8	0.1	2	1.23	56.63
	hc	0.3	0.55	0.72	-20384.7	18	18	0.12	3	2	53.74
	mmhc	0.23	0.65	0.81	-20673.69	9	6	0.17	2	1.33	57
	aracne	0.26	0.59	0.77	-20355.23	11	9	0.16	2	1.64	55.69
0.0625	pc.stable	0.28	0.59	0.73	-20665.48	13	8	0.1	2	1.23	56.63
	hc	0.29	0.54	0.72	-20334.93	18	17	0.11	3	1.89	53.96
	mmhc	0.21	0.72	0.84	-20736.97	9	6	0.17	2	1.33	56.98
	aracne	0.26	0.58	0.77	-20355.23	11	9	0.16	2	1.64	55.69
0.125	pc.stable	0.26	0.61	0.76	-20694.27	11	7	0.13	2	1.27	56.78
	hc	0.29	0.54	0.72	-20334.93	18	17	0.11	3	1.89	53.96
	mmhc	0.23	0.65	0.81	-20673.69	9	6	0.17	2	1.33	57
	aracne	0.26	0.59	0.77	-20355.23	11	9	0.16	2	1.64	55.69
0.25	pc.stable	0.26	0.61	0.76	-20694.27	11	7	0.13	2	1.27	56.78
	hc	0.29	0.56	0.72	-20272.22	18	16	0.1	3	1.78	54.07
	mmhc	0.23	0.65	0.81	-20673.69	9	6	0.17	2	1.33	57
	aracne	0.28	0.6	0.76	-20355.23	11	9	0.16	3	2.18	55.69

0.5	pc.stable	0.26	0.63	0.74	-20749.06	11	7	0.13	2	1.27	56.96
	hc	0.28	0.55	0.71	-20157.91	16	11	0.09	3	1.38	55.19
	mmhc	0.21	0.64	0.83	-20717.62	9	5	0.14	2	1.11	57.2
	aracne	0.28	0.61	0.74	-20335.88	11	8	0.15	3	2.91	55.91
1	pc.stable	0.26	0.62	0.77	-20910.79	10	6	0.13	2	1.4	57.49
	hc	0.27	0.56	0.72	-20186.7	14	10	0.11	3	1.43	55.34
	mmhc	0.2	0.67	0.85	-20879.35	8	4	0.14	1	1	57.73
	aracne	0.28	0.61	0.74	-20335.88	11	8	0.15	3	2.91	55.91
2	pc.stable	0.27	0.59	0.74	-20958.81	9	5	0.14	2	1.33	57.87
	hc	0.27	0.56	0.74	-20229.39	12	9	0.14	3	1.5	55.54
	mmhc	0.2	0.67	0.85	-20879.35	8	4	0.14	1	1	57.73
	aracne	0.28	0.61	0.74	-20335.88	11	8	0.15	3	2.91	55.91
4	pc.stable	0.26	0.6	0.79	-21001.51	7	4	0.19	2	1.43	58.07
	hc	0.25	0.55	0.89	-21202	4	2	0.33	1	2	58.78
	mmhc	0.25	0.55	0.89	-21202	4	2	0.33	1	2	58.78
	aracne	0.25	0.55	0.89	-21202	4	2	0.33	1	2	58.78

Table 39

		$\mathbf{X}_{T_w}$- PAM-B_{causal}. 4 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.33	0.4	0.62	-21488.54	15	9	0.09	2	1.2	56.58
	hc	0.36	0.45	0.64	-22028.33	19	19	0.11	3	2	53.72
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	57
	aracne	NA	NA	NaN	-21779.07	11	9	0.16	2	1.64	55.69
0.015625	pc.stable	0.35	0.34	0.62	-21602.49	13	8	0.1	2	1.23	56.65
	hc	0.36	0.46	0.64	-22028.33	19	19	0.11	3	2	53.72
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	57
	aracne	NA	NA	NaN	-21779.07	11	9	0.16	2	1.64	55.69
0.03125	pc.stable	0.32	0.5	0.62	-21521.09	13	8	0.1	2	1.23	56.63
	hc	0.35	0.45	0.65	-21342.39	18	18	0.12	3	2	53.74
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	57
	aracne	NA	NA	NaN	-21779.07	11	9	0.16	2	1.64	55.69
0.0625	pc.stable	0.32	0.5	0.62	-21521.09	13	8	0.1	2	1.23	56.63
	hc	0.37	0.44	0.62	-21355.47	18	17	0.11	3	1.89	53.96
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	56.98
	aracne	NA	NA	NaN	-21779.07	11	9	0.16	2	1.64	55.69
0.125	pc.stable	0.26	0.64	0.66	-21536.26	11	7	0.13	2	1.27	56.78
	hc	0.34	0.48	0.66	-21296.88	18	17	0.11	3	1.89	53.96
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	57
	aracne	NA	NA	NaN	-21779.07	11	9	0.16	2	1.64	55.69
0.25	pc.stable	0.26	0.64	0.66	-21536.26	11	7	0.13	2	1.27	56.78
	hc	0.36	0.45	0.65	-21358.11	18	16	0.1	3	1.78	54.07
	mmhc	NA	NA	NaN	-21779.07	9	6	0.17	2	1.33	57

	aracne	NA	NA	NaN	-21779.07	11	9	0.16	3	2.18	55.69
0.5	pc.stable	0.26	0.65	0.68	-21619.11	11	7	0.13	2	1.27	56.96
	hc	0.34	0.46	0.62	-21236.85	16	11	0.09	3	1.38	55.19
	mmhc	NA	NA	NaN	-21779.07	9	5	0.14	2	1.11	57.2
	aracne	NA	NA	NaN	-21779.07	11	8	0.15	3	2.91	55.91
1	pc.stable	0.28	0.56	0.69	-21515.47	10	6	0.13	2	1.4	57.49
	hc	0.34	0.45	0.62	-21265.64	14	10	0.11	3	1.43	55.34
	mmhc	NA	NA	NaN	-21779.07	8	4	0.14	1	1	57.73
	aracne	NA	NA	NaN	-21779.07	11	8	0.15	3	2.91	55.91
2	pc.stable	0.28	0.56	0.69	-21515.47	9	5	0.14	2	1.33	57.87
	hc	0.32	0.48	0.69	-21448.75	12	9	0.14	3	1.5	55.54
	mmhc	NA	NA	NaN	-21779.07	8	4	0.14	1	1	57.73
	aracne	NA	NA	NaN	-21779.07	11	8	0.15	3	2.91	55.91
4	pc.stable	0.26	0.58	0.75	-21558.17	7	4	0.19	2	1.43	58.07
	hc	NA	NA	NaN	-21779.07	4	2	0.33	1	2	58.78
	mmhc	NA	NA	NaN	-21779.07	4	2	0.33	1	2	58.78
	aracne	NA	NA	NaN	-21779.07	4	2	0.33	1	2	58.78

Table 40

		$\mathbf{X}_{T_d}$- PAM-B_{co}. 4 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.2	0.7	0.8	-27942.28	25	20	0.07	4	1.6	72.77
	hc	0.21	0.7	0.81	-27326.26	22	19	0.08	3	1.73	72.19
	mmhc	0.19	0.75	0.8	-27500.26	21	15	0.07	4	1.43	73.01
	aracne	0.2	0.71	0.84	-27855.13	7	5	0.24	2	1.43	75.54
0.015625	pc.stable	0.19	0.73	0.8	-27881.37	23	18	0.07	4	1.57	72.83
	hc	0.21	0.7	0.81	-27326.26	22	19	0.08	3	1.73	72.19
	mmhc	0.19	0.76	0.8	-27500.26	21	15	0.07	4	1.43	73.01
	aracne	0.2	0.71	0.84	-27855.13	7	5	0.24	2	1.43	75.54
0.03125	pc.stable	0.18	0.76	0.82	-27829.4	22	16	0.07	4	1.45	72.96
	hc	0.2	0.72	0.83	-27357.42	22	18	0.08	3	1.64	72.3
	mmhc	0.19	0.74	0.81	-27452.36	18	13	0.08	4	1.44	73.26
	aracne	0.2	0.71	0.84	-27855.13	7	5	0.24	2	1.43	75.54
0.0625	pc.stable	0.18	0.76	0.82	-27829.4	22	16	0.07	4	1.45	72.96
	hc	0.21	0.68	0.8	-27214.93	22	16	0.07	3	1.45	72.43
	mmhc	0.19	0.74	0.81	-27401.42	17	12	0.09	4	1.41	73.33
	aracne	0.2	0.71	0.84	-27855.13	7	5	0.24	2	1.43	75.54
0.125	pc.stable	0.18	0.76	0.82	-27829.4	22	16	0.07	4	1.45	72.96
	hc	0.21	0.7	0.8	-27184.06	21	16	0.08	3	1.52	72.34
	mmhc	0.18	0.77	0.8	-27371.5	15	10	0.1	4	1.33	73.75
	aracne	0.2	0.71	0.84	-27855.13	7	5	0.24	2	1.43	75.54
0.25	pc.stable	0.18	0.76	0.82	-27829.4	22	16	0.07	4	1.45	72.96
	hc	0.21	0.7	0.79	-27067.5	20	14	0.07	3	1.4	72.58
	mmhc	0.17	0.75	0.81	-27432.56	14	8	0.09	2	1.14	74.16

	aracne	0.25	0.59	0.84	-27855.13	7	5	0.24	3	2.86	75.54
0.5	pc.stable	0.2	0.73	0.8	-27708.14	22	18	0.08	4	1.64	72.49
	hc	0.21	0.7	0.78	-27027.8	19	13	0.08	3	1.37	72.68
	mmhc	0.18	0.76	0.81	-27432.56	14	8	0.09	2	1.14	74.16
	aracne	0.25	0.59	0.84	-27855.13	7	5	0.24	3	2.86	75.54
1	pc.stable	0.2	0.73	0.79	-27689.91	21	18	0.09	4	1.71	72.46
	hc	0.2	0.72	0.78	-27040.92	18	12	0.08	3	1.33	72.79
	mmhc	0.2	0.69	0.82	-27468.51	12	7	0.11	3	1.67	74.33
	aracne	0.25	0.59	0.84	-27855.13	7	5	0.24	3	2.86	75.54
2	pc.stable	0.2	0.73	0.78	-27650.21	21	17	0.08	4	1.62	72.56
	hc	0.2	0.72	0.8	-27127.39	15	10	0.1	3	1.33	73.18
	mmhc	0.2	0.69	0.82	-27468.51	12	7	0.11	3	1.67	74.33
	aracne	0.24	0.6	0.87	-27905.66	6	4	0.27	3	2.67	75.75
4	pc.stable	0.2	0.72	0.8	-27294.26	20	16	0.08	4	1.6	72.62
	hc	0.28	0.59	0.83	-28444.45	2	1	1	1	2	77.45
	mmhc	0.28	0.59	0.83	-28444.45	2	1	1	1	2	77.45
	aracne	0.28	0.59	0.83	-28444.45	2	1	1	1	2	77.45

Table 41

		$\mathbf{X}_{T_d}$- PAM-B_{causal}. 4 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.22	0.69	0.76	-27990.42	24	16	0.06	1	1.33	73.77
	hc	0.24	0.66	0.76	-27917.65	30	21	0.05	1	1.4	73.5
	mmhc	0.22	0.68	0.74	-27798.34	20	12	0.06	1	1.2	74.55
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.015625	pc.stable	0.22	0.69	0.77	-28112.72	23	15	0.06	1	1.3	74.18
	hc	0.24	0.65	0.76	-27917.65	30	21	0.05	1	1.4	73.5
	mmhc	0.21	0.72	0.75	-27894.07	17	10	0.07	1	1.18	74.96
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.03125	pc.stable	0.22	0.69	0.77	-28112.72	23	15	0.06	1	1.3	74.18
	hc	0.24	0.66	0.76	-27850.34	29	20	0.05	1	1.38	73.62
	mmhc	0.21	0.72	0.75	-27894.07	17	10	0.07	1	1.18	74.96
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.0625	pc.stable	0.22	0.69	0.77	-28112.72	23	15	0.06	1	1.3	74.18
	hc	0.24	0.64	0.75	-27839.05	29	19	0.05	1	1.31	73.75
	mmhc	0.19	0.75	0.76	-27974.63	15	9	0.09	1	1.2	75.25
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.125	pc.stable	0.22	0.69	0.77	-28112.72	23	15	0.06	1	1.3	74.18
	hc	0.24	0.65	0.75	-27808.18	28	19	0.05	1	1.36	73.66
	mmhc	0.19	0.77	0.76	-28063.8	12	7	0.11	1	1.17	75.87
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.25	pc.stable	0.22	0.69	0.76	-27990.42	24	16	0.06	1	1.33	73.77
	hc	0.24	0.66	0.74	-27671.07	21	13	0.06	1	1.24	74.31

	mmhc	0.19	0.76	0.74	-28141.29	10	5	0.11	1	1	76.32
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
0.5	pc.stable	0.22	0.68	0.76	-27990.42	24	16	0.06	1	1.33	73.77
	hc	0.23	0.67	0.73	-27623.88	19	11	0.06	1	1.16	74.46
	mmhc	0.17	0.79	0.75	-28261.44	8	4	0.14	1	1	76.73
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
1	pc.stable	0.22	0.7	0.76	-28152.89	20	14	0.07	1	1.4	74.37
	hc	0.23	0.69	0.74	-27655.05	18	10	0.07	1	1.11	74.62
	mmhc	0.17	0.79	0.75	-28261.44	8	4	0.14	1	1	76.73
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
2	pc.stable	0.22	0.69	0.76	-28152.89	20	14	0.07	1	1.4	74.37
	hc	0.21	0.71	0.75	-27937.99	11	6	0.11	1	1.09	75.69
	mmhc	0.17	0.78	0.76	-28297.39	6	3	0.2	1	1	76.9
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
4	pc.stable	0.22	0.69	0.76	-28152.89	20	14	0.07	1	1.4	74.37
	hc	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
	mmhc	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27
	aracne	NA	NA	NaN	-28718.1	0	0	NaN	0	NaN	78.27

Table 42

		$\mathbf{X}_{T_w}$- PAM-B_{co}. 5 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_l$$ength$	degree	entropy
0.0078125	pc.stable	0.26	0.62	0.79	-24328.79	12	8	0.12	2	1.33	65.83
	hc	0.29	0.53	0.72	-25524.81	21	18	0.09	3	1.71	62.55
	mmhc	0.24	0.68	0.82	-24342.62	10	7	0.16	2	1.4	66.01
	aracne	0.28	0.55	0.74	-23739.27	11	9	0.16	5	1.64	64.57
0.015625	pc.stable	0.25	0.62	0.79	-24453.75	12	7	0.11	1	1.17	66.32
	hc	0.29	0.53	0.72	-25524.81	21	18	0.09	3	1.71	62.55
	mmhc	0.2	0.77	0.87	-24233.45	8	5	0.18	2	1.25	66.48
	aracne	0.28	0.55	0.74	-23739.27	11	9	0.16	5	1.64	64.57
0.03125	pc.stable	0.25	0.63	0.79	-24453.75	12	7	0.11	1	1.17	66.32
	hc	0.29	0.53	0.72	-25524.81	21	18	0.09	3	1.71	62.55
	mmhc	0.18	0.81	0.9	-24358.41	8	4	0.14	1	1	66.96
	aracne	0.28	0.55	0.74	-23739.27	11	9	0.16	5	1.64	64.57
0.0625	pc.stable	0.27	0.61	0.76	-24514.5	12	7	0.11	2	1.17	66.54
	hc	0.3	0.54	0.73	-24467.64	20	17	0.09	3	1.7	62.66
	mmhc	0.18	0.81	0.9	-24358.41	8	4	0.14	1	1	66.96
	aracne	0.28	0.55	0.74	-23739.27	11	9	0.16	5	1.64	64.57
0.125	pc.stable	0.27	0.61	0.76	-24514.5	12	7	0.11	2	1.17	66.54
	hc	0.3	0.54	0.74	-24286.48	19	16	0.09	3	1.68	62.71
	mmhc	0.18	0.81	0.9	-24358.41	8	4	0.14	1	1	66.96
	aracne	0.28	0.53	0.74	-23739.27	11	9	0.16	5	1.82	64.57
0.25	pc.stable	0.24	0.66	0.8	-24528.33	10	6	0.13	2	1.2	66.71
	hc	0.29	0.54	0.73	-23687.86	14	10	0.11	3	1.43	64.3

	mmhc	0.21	0.7	0.88	-24358.41	8	4	0.14	1	1	66.96
	aracne	0.29	0.55	0.74	-23739.27	11	9	0.16	5	3.27	64.57
0.5	pc.stable	0.24	0.67	0.8	-24528.33	10	6	0.13	2	1.2	66.71
	hc	0.29	0.54	0.73	-23687.86	14	10	0.11	3	1.43	64.3
	mmhc	0.21	0.7	0.88	-24358.41	8	4	0.14	1	1	66.96
	aracne	0.29	0.55	0.74	-23739.27	11	9	0.16	5	3.27	64.57
1	pc.stable	0.24	0.67	0.8	-24528.33	10	6	0.13	2	1.2	66.71
	hc	0.28	0.55	0.74	-23722.72	12	9	0.14	3	1.5	64.53
	mmhc	0.28	0.51	0.88	-24358.41	8	4	0.14	1	1.5	66.96
	aracne	0.29	0.55	0.74	-23739.27	11	9	0.16	5	3.27	64.57
2	pc.stable	0.24	0.66	0.76	-24405.77	10	5	0.11	1	1	66.96
	hc	0.27	0.55	0.92	-24515.9	6	3	0.2	1	2	67.54
	mmhc	0.27	0.55	0.92	-24515.9	6	3	0.2	1	2	67.54
	aracne	0.27	0.55	0.92	-24515.9	6	3	0.2	1	2	67.54
4	pc.stable	0.24	0.66	0.76	-24405.77	10	5	0.11	1	1	66.96
	hc	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01
	mmhc	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01
	aracne	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01

Table 43

		$\mathbf{X}_{T_w}$- PAM-B_{causal}. 5 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.36	0.37	0.61	-25078.04	12	8	0.12	2	1.33	65.83
	hc	0.4	0.35	0.57	-26371.6	21	18	0.09	3	1.71	62.55
	mmhc	0.36	0.44	0.58	-25223.87	10	7	0.16	2	1.4	66.01
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	1.64	64.57
0.015625	pc.stable	0.36	0.38	0.63	-25114.83	12	7	0.11	1	1.17	66.32
	hc	0.4	0.33	0.57	-26371.6	21	18	0.09	3	1.71	62.55
	mmhc	NA	NA	NaN	-25258.73	8	5	0.18	2	1.25	66.48
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	1.64	64.57
0.03125	pc.stable	0.36	0.38	0.63	-25114.83	12	7	0.11	1	1.17	66.32
	hc	0.41	0.33	0.56	-26247.98	21	18	0.09	3	1.71	62.55
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	1.64	64.57
0.0625	pc.stable	0.36	0.38	0.63	-25114.83	12	7	0.11	2	1.17	66.54
	hc	0.4	0.37	0.6	-25188.68	20	17	0.09	3	1.7	62.66
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	1.64	64.57
0.125	pc.stable	0.36	0.38	0.63	-25114.83	12	7	0.11	2	1.17	66.54
	hc	0.4	0.36	0.6	-25188.68	19	16	0.09	3	1.68	62.71
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	1.82	64.57
	pc.stable	0.32	0.46	0.67	-25007.47	10	6	0.13	2	1.2	66.71

0.25

	hc	0.37	0.36	0.62	-24723.9	14	10	0.11	3	1.43	64.3
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	3.27	64.57
0.5	pc.stable	0.31	0.5	0.69	-25021.31	10	6	0.13	2	1.2	66.71
	hc	0.37	0.36	0.62	-24723.9	14	10	0.11	3	1.43	64.3
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	3.27	64.57
1	pc.stable	0.31	0.5	0.69	-25021.31	10	6	0.13	2	1.2	66.71
	hc	0.36	0.36	0.65	-24820.17	12	9	0.14	3	1.5	64.53
	mmhc	NA	NA	NaN	-25258.73	8	4	0.14	1	1.5	66.96
	aracne	NA	NA	NaN	-25258.73	11	9	0.16	5	3.27	64.57
2	pc.stable	0.32	0.49	0.65	-24932.86	10	5	0.11	1	1	66.96
	hc	NA	NA	NaN	-25258.73	6	3	0.2	1	2	67.54
	mmhc	NA	NA	NaN	-25258.73	6	3	0.2	1	2	67.54
	aracne	NA	NA	NaN	-25258.73	6	3	0.2	1	2	67.54
4	pc.stable	0.32	0.49	0.65	-24932.86	10	5	0.11	1	1	66.96
	hc	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01
	mmhc	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01
	aracne	NA	NA	NaN	-25258.73	0	0	NaN	0	NaN	70.01

Table 44

		$\mathbf{X}_{T_d}$- PAM-B_{co}. 5 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_{l}ength$	degree	entropy
0.0078125	pc.stable	0.26	0.64	0.77	-34824.05	23	20	0.08	4	1.74	83.46
	hc	0.26	0.63	0.76	-32193.59	22	17	0.07	4	1.55	83.28
	mmhc	0.25	0.65	0.77	-31996.88	21	15	0.07	4	1.43	83.58
	aracne	0.24	0.63	0.82	-32384.16	7	5	0.24	2	1.43	87.33
0.015625	pc.stable	0.26	0.62	0.77	-32384.96	22	16	0.07	4	1.45	83.6
	hc	0.26	0.63	0.76	-32014.6	21	16	0.08	4	1.52	83.49
	mmhc	0.25	0.64	0.76	-31884.89	20	14	0.07	4	1.4	83.9
	aracne	0.24	0.63	0.82	-32384.16	7	5	0.24	2	1.43	87.33
0.03125	pc.stable	0.26	0.61	0.77	-32272.97	21	15	0.07	4	1.43	83.92
	hc	0.26	0.63	0.76	-32014.6	21	16	0.08	4	1.52	83.49
	mmhc	0.25	0.64	0.76	-31884.89	20	14	0.07	4	1.4	83.9
	aracne	0.24	0.64	0.82	-32384.16	7	5	0.24	2	1.43	87.33
0.0625	pc.stable	0.26	0.61	0.77	-32272.97	21	15	0.07	4	1.43	83.92
	hc	0.26	0.63	0.76	-31863.21	21	15	0.07	3	1.43	83.67
	mmhc	0.23	0.66	0.78	-31862.71	17	11	0.08	3	1.29	84.73
	aracne	0.24	0.64	0.82	-32384.16	7	5	0.24	2	1.43	87.33
0.125	pc.stable	0.26	0.62	0.77	-32272.97	21	15	0.07	4	1.43	83.92
	hc	0.26	0.62	0.76	-31849.67	21	15	0.07	3	1.43	83.69
	mmhc	0.24	0.65	0.77	-31676.55	17	10	0.07	2	1.18	84.74
	aracne	0.24	0.64	0.82	-32384.16	7	5	0.24	2	1.43	87.33
	pc.stable	0.25	0.63	0.77	-32103.62	20	14	0.07	4	1.4	83.96

0.25

	hc	0.25	0.65	0.77	-31656.46	18	12	0.08	3	1.33	84.43
	mmhc	0.24	0.66	0.77	-31676.55	17	10	0.07	2	1.18	84.74
	aracne	0.27	0.56	0.82	-32384.16	7	5	0.24	3	2.86	87.33
0.5	pc.stable	0.25	0.63	0.77	-31917.46	20	13	0.07	4	1.3	83.97
	hc	0.23	0.68	0.78	-31647.95	17	11	0.08	3	1.29	84.53
	mmhc	0.24	0.64	0.77	-31815.35	15	9	0.09	2	1.2	85.25
	aracne	0.27	0.56	0.82	-32384.16	7	5	0.24	3	2.86	87.33
1	pc.stable	0.25	0.63	0.77	-31754.63	19	12	0.07	4	1.26	84.15
	hc	0.24	0.67	0.8	-31700.71	14	9	0.1	3	1.29	84.93
	mmhc	0.27	0.58	0.78	-31837.03	13	8	0.1	3	1.69	85.44
	aracne	0.27	0.57	0.84	-32415.24	6	4	0.27	3	2.67	87.54
2	pc.stable	0.25	0.65	0.76	-31732.94	20	13	0.07	4	1.3	83.96
	hc	0.28	0.57	0.81	-32166.86	10	5	0.11	1	1.6	86.73
	mmhc	0.28	0.57	0.81	-32166.86	10	5	0.11	1	1.6	86.73
	aracne	0.29	0.54	0.85	-32522.22	6	3	0.2	1	2	87.97
4	pc.stable	0.25	0.64	0.76	-31535.9	19	12	0.07	4	1.26	84.09
	hc	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
	mmhc	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29

Table 45

		$\mathbf{X}_{T_d}$- PAM-B_{causal}. 5 levels discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entropy
0.0078125	pc.stable	0.28	0.6	0.73	-35874.75	24	18	0.07	1	1.5	85.13
	hc	0.29	0.57	0.71	-33727.46	26	17	0.05	1	1.31	85.14
	mmhc	0.27	0.61	0.71	-32460.9	17	10	0.07	1	1.18	86.5
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
0.015625	pc.stable	0.28	0.58	0.73	-33633.08	22	15	0.06	1	1.36	85.22
	hc	0.28	0.58	0.71	-33538.57	25	16	0.05	1	1.28	85.28
	mmhc	0.27	0.61	0.71	-32460.9	17	10	0.07	1	1.18	86.5
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
0.03125	pc.stable	0.29	0.57	0.72	-33521.09	21	14	0.07	1	1.33	85.54
	hc	0.28	0.58	0.71	-32413.94	25	15	0.05	1	1.2	85.32
	mmhc	0.26	0.61	0.73	-32482.57	16	9	0.07	1	1.12	86.69
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
0.0625	pc.stable	0.28	0.57	0.72	-32381.63	21	13	0.06	1	1.24	85.52
	hc	0.28	0.58	0.71	-32215.54	24	14	0.05	1	1.17	85.31
	mmhc	0.26	0.63	0.73	-32554.85	14	8	0.09	1	1.14	87.02
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
0.125	pc.stable	0.28	0.56	0.71	-32489.54	20	12	0.06	1	1.2	85.94
	hc	0.28	0.58	0.71	-32029.38	24	13	0.05	1	1.08	85.32
	mmhc	0.26	0.63	0.73	-32368.7	14	7	0.08	1	1	87.03
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29

0.25	pc.stable	0.28	0.56	0.71	-32489.54	20	12	0.06	1	1.2	85.94
	hc	0.26	0.62	0.72	-32114.83	20	11	0.06	1	1.1	85.81
	mmhc	0.27	0.6	0.72	-32507.5	12	6	0.09	1	1	87.54
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
0.5	pc.stable	0.28	0.56	0.71	-32489.54	20	12	0.06	1	1.2	85.94
	hc	0.26	0.62	0.72	-32114.83	20	11	0.06	1	1.1	85.81
	mmhc	0.27	0.62	0.72	-32619.96	10	5	0.11	1	1	87.97
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
1	pc.stable	0.28	0.56	0.71	-32489.54	20	12	0.06	1	1.2	85.94
	hc	0.27	0.62	0.74	-32210.71	15	8	0.08	1	1.07	86.46
	mmhc	0.29	0.56	0.72	-32529.18	10	5	0.11	1	1	87.72
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
2	pc.stable	0.27	0.58	0.71	-32467.86	21	13	0.06	1	1.24	85.76
	hc	0.25	0.67	0.76	-32871.01	4	2	0.33	1	1	89.05
	mmhc	0.25	0.67	0.76	-32871.01	4	2	0.33	1	1	89.05
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
4	pc.stable	0.27	0.58	0.7	-32520.4	19	12	0.07	1	1.26	86.03
	hc	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
	mmhc	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29
	aracne	NA	NA	NaN	-33226.37	0	0	NaN	0	NaN	90.29

Table 46

		$\mathbf{X}_{T_w}$- PAM-B_{co}. dynamic 3-5 discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_length$	degree	entropy
0.0078125	pc.stable	0.26	0.58	0.72	-18440.4	15	11	0.1	4	1.47	50.31
	hc	0.25	0.62	0.76	-18617.33	18	17	0.11	2	1.89	49.22
	mmhc	0.29	0.55	0.65	-18760.43	11	7	0.13	4	1.27	51.6
	aracne	0.26	0.59	0.74	-18344.65	12	11	0.17	4	2	49.97
0.015625	pc.stable	0.29	0.57	0.69	-18685.98	15	9	0.09	2	1.2	51.06
	hc	0.26	0.6	0.76	-18520.44	18	16	0.1	2	1.78	49.45
	mmhc	0.27	0.6	0.69	-18841.89	11	6	0.11	2	1.09	51.9
	aracne	0.26	0.59	0.74	-18344.65	12	11	0.17	4	2	49.97
0.03125	pc.stable	0.28	0.57	0.7	-18804.29	12	7	0.11	1	1.17	51.59
	hc	0.25	0.61	0.75	-18842.36	19	16	0.09	2	1.68	49.56
	mmhc	0.27	0.6	0.69	-18841.89	11	6	0.11	2	1.09	51.9
	aracne	0.26	0.59	0.74	-18344.65	12	11	0.17	4	2	49.97
0.0625	pc.stable	0.28	0.57	0.7	-18804.29	12	7	0.11	1	1.17	51.59
	hc	0.25	0.62	0.75	-18842.36	19	16	0.09	2	1.68	49.56
	mmhc	0.28	0.59	0.67	-18841.89	11	6	0.11	2	1.09	51.9
	aracne	0.26	0.59	0.74	-18344.65	12	11	0.17	4	2	49.97
0.125	pc.stable	0.28	0.61	0.68	-18958.09	11	6	0.11	1	1.09	52.12
	hc	0.24	0.63	0.75	-18175.68	18	14	0.09	3	1.56	49.46
	mmhc	0.26	0.61	0.69	-18931.64	10	5	0.11	1	1	52.28

	aracne	0.26	0.57	0.75	-18344.65	12	11	0.17	4	2.17	49.97
	pc.stable	0.28	0.61	0.68	-18958.09	11	6	0.11	1	1.09	52.12
	hc	0.27	0.6	0.72	-18340.18	16	12	0.1	3	1.5	50.09
0.25	mmhc	0.23	0.67	0.76	-18966.16	8	4	0.14	1	1	52.44
	aracne	0.26	0.58	0.77	-18299.85	12	10	0.15	3	2.33	50.1
	pc.stable	0.26	0.65	0.73	-18992.61	9	5	0.14	1	1.11	52.28
	hc	0.24	0.64	0.74	-18305.15	14	10	0.11	3	1.43	50.25
0.5	mmhc	0.24	0.65	0.73	-18966.16	8	4	0.14	1	1	52.44
	aracne	0.26	0.58	0.77	-18299.85	12	10	0.15	5	3.33	50.1
	pc.stable	0.24	0.65	0.73	-18966.16	8	4	0.14	1	1	52.44
	hc	0.25	0.63	0.71	-18175.89	12	11	0.17	7	1.83	49.88
1	mmhc	0.24	0.65	0.73	-18966.16	8	4	0.14	1	1	52.44
	aracne	0.26	0.58	0.77	-18299.85	12	10	0.15	5	3.33	50.1
	pc.stable	0.24	0.65	0.74	-18966.16	8	4	0.14	1	1	52.44
	hc	0.24	0.64	0.71	-18252.08	12	10	0.15	5	1.67	50.16
2	mmhc	0.24	0.66	0.74	-18838.49	9	5	0.14	2	1.56	52.05
	aracne	0.26	0.58	0.78	-18437.65	11	8	0.15	3	2.91	50.68
	pc.stable	0.24	0.65	0.74	-18966.16	8	4	0.14	1	1	52.44
	hc	0.24	0.66	0.73	-18901.86	8	5	0.18	2	2	52.29
4	mmhc	0.23	0.69	0.75	-19001.93	7	4	0.19	2	1.71	52.61
	aracne	0.26	0.6	0.76	-19129.59	6	3	0.2	1	2	53

Table 47

		$\mathbf{X}_{T_w}$- PAM-B_{causal}. dynamic 3-5 discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	max$_l ength$	degree	entropy
	pc.stable	0.3	0.52	0.63	-19409.41	15	11	0.1	4	1.47	50.31
	hc	0.34	0.51	0.66	-21230.19	18	17	0.11	2	1.89	49.22
0.0078125	mmhc	0.35	0.41	0.5	-19675.1	11	7	0.13	4	1.27	51.6
	aracne	NA	NA	NaN	-19709.63	12	11	0.17	4	2	49.97
	pc.stable	0.3	0.51	0.64	-19509.48	15	9	0.09	2	1.2	51.06
	hc	0.33	0.55	0.67	-19848.8	18	16	0.1	2	1.78	49.45
0.015625	mmhc	0.35	0.41	0.5	-19675.1	11	6	0.11	2	1.09	51.9
	aracne	NA	NA	NaN	-19709.63	12	11	0.17	4	2	49.97
	pc.stable	0.36	0.37	0.56	-19605.64	12	7	0.11	1	1.17	51.59
	hc	0.31	0.57	0.7	-19757.05	19	16	0.09	2	1.68	49.56
0.03125	mmhc	0.35	0.41	0.5	-19675.1	11	6	0.11	2	1.09	51.9
	aracne	NA	NA	NaN	-19709.63	12	11	0.17	4	2	49.97
	pc.stable	0.3	0.49	0.64	-19540.19	12	7	0.11	1	1.17	51.59
	hc	0.31	0.55	0.69	-19445.57	19	16	0.09	2	1.68	49.56
0.0625	mmhc	0.35	0.41	0.5	-19675.1	11	6	0.11	2	1.09	51.9
	aracne	NA	NA	NaN	-19709.63	12	11	0.17	4	2	49.97
	pc.stable	0.3	0.49	0.64	-19540.19	11	6	0.11	1	1.09	52.12
	hc	0.29	0.57	0.66	-19100.07	18	14	0.09	3	1.56	49.46
0.125	mmhc	0.35	0.41	0.5	-19675.1	10	5	0.11	1	1	52.28

	aracne	NA	NA	NaN	-19709.63	12	11	0.17	4	2.17	49.97
0.25	pc.stable	0.36	0.37	0.56	-19605.64	11	6	0.11	1	1.09	52.12
	hc	0.33	0.52	0.63	-19243.67	16	12	0.1	3	1.5	50.09
	mmhc	NA	NA	NaN	-19709.63	8	4	0.14	1	1	52.44
	aracne	NA	NA	NaN	-19709.63	12	10	0.15	3	2.33	50.1
0.5	pc.stable	0.37	0.34	0.62	-19640.16	9	5	0.14	1	1.11	52.28
	hc	0.32	0.51	0.61	-19014.54	14	10	0.11	3	1.43	50.25
	mmhc	NA	NA	NaN	-19709.63	8	4	0.14	1	1	52.44
	aracne	NA	NA	NaN	-19709.63	12	10	0.15	5	3.33	50.1
1	pc.stable	0.37	0.34	0.62	-19640.16	8	4	0.14	1	1	52.44
	hc	0.32	0.51	0.61	-19112.43	12	11	0.17	7	1.83	49.88
	mmhc	NA	NA	NaN	-19709.63	8	4	0.14	1	1	52.44
	aracne	NA	NA	NaN	-19709.63	12	10	0.15	5	3.33	50.1
2	pc.stable	NA	NA	NaN	-19709.63	8	4	0.14	1	1	52.44
	hc	0.29	0.59	0.66	-19275.56	12	10	0.15	5	1.67	50.16
	mmhc	NA	NA	NaN	-19709.63	9	5	0.14	2	1.56	52.05
	aracne	NA	NA	NaN	-19709.63	11	8	0.15	3	2.91	50.68
4	pc.stable	0.26	0.68	0.63	-19627.12	8	4	0.14	1	1	52.44
	hc	0.3	0.55	0.6	-19609.56	8	5	0.18	2	2	52.29
	mmhc	NA	NA	NaN	-19709.63	7	4	0.19	2	1.71	52.61
	aracne	NA	NA	NaN	-19709.63	6	3	0.2	1	2	53

Table 48

		$\mathbf{X}_{T_d}$- PAM-B_{co}. dynamic 3-5 discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_length$	degree	entropy
0.0078125	pc.stable	0.18	0.76	0.78	-25439.49	20	18	0.09	3	1.8	66.04
	hc	0.18	0.76	0.79	-25475.29	24	20	0.07	3	1.67	66.06
	mmhc	0.17	0.78	0.79	-25269.66	20	15	0.08	3	1.5	66.78
	aracne	0.18	0.76	0.8	-25440.49	8	7	0.25	6	2	68.87
0.015625	pc.stable	0.18	0.76	0.83	-24972.72	19	15	0.09	4	1.58	66.67
	hc	0.19	0.75	0.79	-25259.73	23	19	0.08	4	1.65	66.14
	mmhc	0.17	0.77	0.8	-25165.06	18	13	0.08	2	1.44	67.08
	aracne	0.15	0.8	0.86	-25521	8	7	0.25	3	1.75	68.89
0.03125	pc.stable	0.17	0.75	0.83	-25194.22	18	13	0.08	3	1.44	67.41
	hc	0.19	0.75	0.78	-25231.13	23	18	0.07	4	1.57	66.24
	mmhc	0.17	0.77	0.81	-25285.68	17	12	0.09	2	1.41	67.49
	aracne	0.15	0.8	0.86	-25521	8	7	0.25	3	1.75	68.89
0.0625	pc.stable	0.18	0.76	0.81	-25188.79	18	12	0.08	2	1.33	67.53
	hc	0.19	0.74	0.78	-24993.62	23	18	0.07	4	1.57	66.29
	mmhc	0.17	0.78	0.8	-25197.25	17	11	0.08	2	1.29	67.51
	aracne	0.15	0.8	0.86	-25521	8	7	0.25	3	1.75	68.89
0.125	pc.stable	0.17	0.78	0.8	-25197.25	17	11	0.08	1	1.29	67.51
	hc	0.18	0.76	0.79	-24927.26	22	16	0.07	3	1.45	66.43

	mmhc	0.16	0.8	0.8	-25117.72	18	11	0.07	2	1.22	67.35
	aracne	0.17	0.77	0.86	-25521	8	7	0.25	3	2	68.89
0.25	pc.stable	0.17	0.78	0.8	-25197.25	17	11	0.08	1	1.29	67.51
	hc	0.16	0.79	0.81	-24969.23	17	12	0.09	4	1.41	67.14
	mmhc	0.16	0.8	0.81	-25193.44	14	8	0.09	2	1.14	67.99
	aracne	0.2	0.7	0.82	-25521	8	7	0.25	3	2.5	68.89
0.5	pc.stable	0.17	0.77	0.8	-25076.63	18	12	0.08	2	1.33	67.11
	hc	0.18	0.74	0.78	-24727.48	18	12	0.08	4	1.33	66.6
	mmhc	0.16	0.79	0.8	-25056.39	16	9	0.07	2	1.12	67.54
	aracne	0.2	0.73	0.78	-25451.67	8	6	0.21	3	3	69
1	pc.stable	0.17	0.77	0.8	-25076.63	18	12	0.08	2	1.33	67.11
	hc	0.18	0.74	0.78	-24727.48	18	12	0.08	4	1.33	66.6
	mmhc	0.17	0.78	0.79	-25080.94	15	8	0.08	2	1.07	67.81
	aracne	0.2	0.73	0.78	-25451.67	8	6	0.21	3	3	69
2	pc.stable	0.17	0.78	0.8	-25197.25	17	11	0.08	1	1.29	67.51
	hc	0.18	0.73	0.77	-24934.94	15	10	0.1	3	1.33	67.41
	mmhc	0.17	0.79	0.79	-25219.74	13	7	0.09	2	1.08	68.32
	aracne	0.21	0.7	0.81	-25491.01	7	5	0.24	3	2.86	69.16
4	pc.stable	0.17	0.78	0.8	-25197.25	17	11	0.08	1	1.29	67.51
	hc	0.14	0.84	0.81	-25628.72	7	4	0.19	2	1.14	69.66
	mmhc	0.13	0.85	0.83	-25730.93	6	3	0.2	1	1	69.98
	aracne	0.2	0.71	0.91	-25866.59	4	2	0.33	1	2	70.38

Table 49

		$\mathbf{X}_{T_d}$- PAM-B_{causal}. dynamic 3-5 discretization									
		brier	accuracy	pr-AUC	BIC	nodes	edges	density	$\max_{l}ength$	degree	entropy
0.0078125	pc.stable	0.2	0.7	0.76	-25486.09	20	16	0.08	1	1.6	67.74
	hc	0.2	0.7	0.74	-25285	23	15	0.06	1	1.3	67.62
	mmhc	0.18	0.75	0.77	-25584.89	15	9	0.09	1	1.2	68.7
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.015625	pc.stable	0.2	0.7	0.75	-25405.32	20	14	0.07	1	1.4	67.78
	hc	0.2	0.7	0.74	-25285	23	15	0.06	1	1.3	67.62
	mmhc	0.18	0.75	0.77	-25584.89	15	9	0.09	1	1.2	68.7
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.03125	pc.stable	0.2	0.69	0.77	-25383.7	19	12	0.07	1	1.26	68.05
	hc	0.19	0.73	0.75	-25328.81	21	15	0.07	1	1.43	67.61
	mmhc	0.18	0.75	0.78	-25705.51	13	8	0.1	1	1.23	69.1
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.0625	pc.stable	0.2	0.72	0.78	-25546.54	20	13	0.07	1	1.3	68.08
	hc	0.19	0.73	0.76	-25318.17	22	15	0.06	1	1.36	67.59
	mmhc	0.18	0.75	0.78	-25705.51	13	8	0.1	1	1.23	69.1
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.125	pc.stable	0.2	0.72	0.76	-25518.84	21	14	0.07	1	1.33	67.94
	hc	0.19	0.73	0.75	-25225.85	22	14	0.06	1	1.27	67.6

	mmhc	0.18	0.75	0.78	-25705.51	13	8	0.1	1	1.23	69.1
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.25	pc.stable	0.2	0.72	0.76	-25518.84	21	14	0.07	1	1.33	67.94
	hc	0.19	0.73	0.74	-25307.37	19	12	0.07	1	1.26	68.14
	mmhc	0.18	0.77	0.75	-25694.2	11	6	0.11	1	1.09	69.41
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
0.5	pc.stable	0.2	0.72	0.76	-25518.84	21	14	0.07	1	1.33	67.94
	hc	0.2	0.7	0.73	-25176.11	18	11	0.07	1	1.22	67.96
	mmhc	0.18	0.77	0.75	-25557.14	13	7	0.09	1	1.08	68.96
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
1	pc.stable	0.2	0.72	0.75	-25262.29	22	14	0.06	1	1.27	67.59
	hc	0.21	0.7	0.72	-25129.18	18	11	0.07	1	1.22	67.8
	mmhc	0.19	0.75	0.73	-25733.3	8	4	0.14	1	1	69.82
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
2	pc.stable	0.2	0.72	0.75	-25382.91	20	13	0.07	1	1.3	68
	hc	0.19	0.74	0.73	-25465.38	13	7	0.09	1	1.08	69.03
	mmhc	0.19	0.74	0.72	-25872.1	6	3	0.2	1	1	70.33
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5
4	pc.stable	0.19	0.73	0.75	-25485.45	18	12	0.08	1	1.33	68.41
	hc	0.16	0.8	0.73	-25904.2	5	3	0.3	1	1.2	70.47
	mmhc	0.11	0.88	0.76	-26108.6	2	1	1	1	1	71.1
	aracne	NA	NA	NaN	-26244.26	0	0	NaN	0	NaN	71.5

Table 50

References

Abramson, B., Brown, J., Edwards, W., Murphy, A., & Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, *12*(1), 57–71.

Agresti, A. (2013). *Categorical data analysis* (3rd ed.). Wiley.

Colombo, D., & Maathuis, M. H. (2015). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, *15*, 3741–3782.

Fuller, W. A. (1996). *Introduction to statistical time series* (2nd ed.). New York: John Wiley and Sons.

Grzegorczyk, M., & Husmeier, D. (n.d.). Non-homogeneous dynamic Bayesian networks for continuous data, journal = Machine Learning, volume = 83, pages = 355–419, year = 2011, doi = 10.1007/s10994-010-5230-7.

Hartemink, A. (2001). *Principled computational methods for the validation and discovery of genetic regulatory networks* (Unpublished doctoral dissertation). Massachusetts Institute of Technology, School of Electrical Engineering and Computer Science.

Hassan, Q. F., Khan, A. u. R., & Madani, S. A. (2018). *Internet of things: Challenges, advances, and applications.* Taylor & Francis, CRC Press. Retrieved from `https://www.books24x7.com/marc.asp?bookid=138737`

Heskes, T., Spanjers, J. J., & Bakker, B. (n.d.). Optimising newspaper sales using neural-Bayesian technology, journal = Neural Computing & Applications, volume = 12, pages = 212–219, year = 2003, doi = 10.1007/s00521-003-0384-x.

Horita, Y., & Yamashita, H. (2019). Bayesian network considering the clustering of the customers in a hair salon. *Cogent Business & Management*, *6*(1). doi: 10.1080/23311975.2019.1641897

Jonckheere, A. R. (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika*, *41*, 133–145. doi: 10.2307/2333011

Kamalabad, M. S., & Grzegorczyk, M. (n.d.). A new Bayesian piecewise linear regression model for dynamic network reconstruction, journal = BMC Bioinformatics, volume = 22, pages = 196, year = 2021, doi = 10.1186/s12859-021-03998-9.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques.* The MIT Press.

Kruppa, J., Liu, Y., Diener, H.-C., Holste, T., Weimar, C., Konig, I. R., & Ziegler, A. (2014). Probability estimation with machine learning methods for dichotomous and multicategory outcome: Applications. *Biometrical Journal*, *56*(4), 564–583.

Kullback, S. (1959). *Information theory and statistics.* New York: John Wiley & Sons. (Reprinted by Dover Publications, 1997)

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., & Califano, A. (2006). Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, *7*(Suppl 1), S7. doi: 10.1186/1471-2105-7-S1-S7

Panda, M., Mousa, A. A. A., & Hassanien, A. E. (2021). Developing an efficient feature engineering and machine learning model for detecting iot-botnet cyber attacks. *IEEE Access*, *9*. doi: 10.1109/ACCESS.2021.3092054

Ragg, T., Menzel, W., Baum, W., & Wigbers, M. (2002). Bayesian learning for sales rate prediction for thousands of retailers. *Neurocomputing*, *43*(1), 127-144. doi: https://doi.org/10.1016/S0925-2312(01)00624-5

Rook, C. S. L. (2025). *Dynamic Bayesian networks application and implementation* (Unpublished doctoral dissertation). University of Groningen, Faculty of Science and Engineering.

Russell, S. J., & Norvig, P. (2022). *Artificial intelligence: A modern approach* (Fourth, Global Edition ed.). Pearson Education Limited.

Salama, K. M., & Freitas, A. A. (2014). Classification with cluster-based Bayesian multi-nets using ant colony optimisation. *Swarm and Evolutionary Computation*, *18*, 54–70. doi: 10.1016/j.swevo.2014.05.001

Schubert, E., & Rousseeuw, P. J. (2021). Fast and eager k-medoids clustering: $o(k)$ runtime improvement of the PAM, CLARA, and CLARANS algorithms. *Information Systems*. doi: https://doi.org/10.1016/j.is.2021.101804

Scutari, M. (n.d.). Bayesian network models for incomplete and dynamic data, journal = Statistica Neerlandica, volume = 74, pages = 397–419, year = 2020, doi = 10.1111/stan.12197.

Scutari, M. (2024). Entropy and the Kullback—Leibler divergence for Bayesian networks: Computational complexity and efficient implementation. *Algorithms*, *17*, 24. doi: 10.3390/a17010024

Scutari, M., & Denis, J.-B. (2021). *Bayesian networks with examples in R* (2nd ed.). Boca Raton, FL: Chapman & Hall/CRC.

Scutari, M., Graafland, C. E., & Gutiérrez, J. M. (n.d.). Who learns better Bayesian network structures: Accuracy and speed of structure learning algorithms, journal = International Journal of Approximate Reasoning, volume = 115, pages = 235–253, year = 2019, doi = https://doi.org/10.1016/j.ijar.2019.10.003.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3/4), 591–611. doi: 10.1093/biomet/52.3-4.591

Terpstra, T. J. (1952). The asymptotic normality and consistency of Kendall's test against trend, when ties are present in one ranking. *Indagationes Mathematicae*, *14*, 327–333.

Tiwari, M., Kang, R., Lee, D., Thrun, S., Piech, C., Shomorony, I., & Zhang, M. J. (2023). Banditpam++: Faster k-medoids clustering. *arXiv preprint arXiv:2310.18844*. Retrieved from `https://arxiv.org/abs/2310.18844`

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003). Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD '03)* (pp. 673–678). Washington, DC, USA: ACM. doi: 10.1145/956750.956838

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*, 31–78. doi: 10.1007/s10994-006-6889-7

Verbraken, T., Verbeke, W., & Baesens, B. (n.d.). Profit optimizing customer churn prediction with Bayesian network classifiers, journal = Intelligent Data Analysis, volume = 18, pages = 3–24, year = 2014, doi = 10.3233/IDA-130625.

Wang, Z., Wang, Q., & Wang, D. W. (2009). Bayesian network based business information retrieval model. *Knowledge and Information Systems*, *20*, 63–79. doi: 10.1007/s10115-008-0151-5

Wit, E., van den Heuvel, E., & Romeijn, J.-W. (2012). All models are wrong...: an introduction to model uncertainty. *Statistica Neerlandica*, *66*(3), 217–236. doi: 10.1111/j.1467-9574.2012.00530.x

Zhao, Z. Y., Xie, M., & West, M. (2016). Dynamic dependence networks: Financial time series forecasting and portfolio decisions. *Applied Stochastic Models in Business and Industry*, *32*, 311–332. doi: 10.1002/asmb.2161