EXDBN: EXACT LEARNING OF DYNAMIC BAYESIAN NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal learning from data has received much attention in recent years. One way of capturing causal relationships is by utilizing Bayesian networks. There, one recovers a weighted directed acyclic graph, in which random variables are represented by vertices, and the weights associated with each edge represent the strengths of the causal relationships between them. This concept is extended to capture dynamic effects by introducing a dependency on past data, which may be captured by the structural equation model, which is utilized in the present contribution to formulate a score-based learning approach. A mixed-integer quadratic program is formulated and an algorithmic solution proposed, in which the pre-generation of exponentially many acyclicity constraints is avoided by utilizing the so-called branch-and-cut ("lazy constraint") method. Comparing the novel approach to the state of the art, we show that the proposed approach turns out to produce excellent results when applied to small and medium-sized synthetic instances of up to 25 time-series. Lastly, two interesting applications in bio-science and finance, to which the method is directly applied, further stress the importance of developing highly accurate (globally convergent) solvers that can handle modest instances.

1 INTRODUCTION

The problem of causal learning using graphical structures has received considerable attention from a wide range of communities in recent years. This attention comes from the wide range of applications including, but not limited to, medicine (Rajapakse & Zhou, 2007), machine learning (Koller & Friedman, 2009), econometrics (Luetkepohl, 2005; Demiralp & Hoover, 2003; Malinsky & Spirtes, 2018) and others (Guo et al., 2020; Assaad et al., 2022). One key reason for this is that in many applications data is abundant, but modeling using first principles may be difficult due to the complexity of the problem at hand (Guo et al., 2020). Some of this complexity may arise due to an abundance of non-linear effects, only a partial ability to observe the system, or unexpected stochastic effects influencing the system. A key benefit of learning via graphical structures is the full explainability of the output; the network may be either used to compute outputs for different situations or the learned graph structure may be inspected and dependencies of particular interest analyzed.

In this contribution, we revisit the score-based learning of dynamic Bayesian networks utilizing a directed acyclic graph (DAG) structure augmented by additional time dependencies from data (Murphy, 2002; Dean & Kanazawa, 1989; Assaad et al., 2022). This approach to learning causality has been successfully applied to a variety of problems, many of which are related to applications in medicine (Zandonà et al., 2019; van Gerven et al., 2008; Michoel & Zhang, 2023; Zhong et al., 2023). Besides medical applications, the dynamic Bayesian network approach representations are widely used in econometrics (Hoover & Demiralp, 2003b) and financial risk modeling (Ballester et al., 2023). This broad scope of applications has spawned a large number of excellent solvers, which under different assumptions are able to discover the underlying causal structure of a system. The use of various assumptions is key to ensure the tractability of a solver, since the the number of constraints that is needed to impose to acyclicity of the representing graph is super-exponential in the number of random variables.

One of the possible assumptions is to separate observational and interventional data (Gao et al., 2022), which reduces the number of dependencies that need to be found. Another is the assumption of underlying continuous dynamics represented by stochastic differential equations Bellot et al.

(2021). One can also assume a-priori knowledge about the time-lagged data and incorporate this knowledge into the solver Sun et al. (2021). One can also deal with the general problem and propose local methods (Pamfil et al., 2020; Gao et al., 2022), which can scale further at the cost of some loss of accuracy. Note that many of the previous works also combine several of these approaches to arrive at solvers that are tractable and applicable to a wide range of applications.

We utilize mixed-integer programming in learning dynamic Bayesian networks. All of the previous works mentioned above focus mostly on solving the curse of dimensionality and scaling with adequate precision. On the other hand, we focus on leveraging fundamental principles that apply to quadratic mixed-integer programs to find global solutions to the score-based DAG learning problem, which results in a high quality reconstruction of the DAG. Furthermore, we tackle the curse of dimensionality by avoiding the pre-generation of the acyclic constraints. It is shown, that given sufficient data, only a small amount of these constraints are actually needed to ensure the acyclicity of the resulting graph, which leads to the runtime generation of these constraints granting a large speedup. The formulation and its implementation are easily reproducible, making it accessible to a wide range of potential practitioners.

2 Problem Formulation

Before formulating the problem of score-based Bayesian network learning as a mixed-integer program, let us describe the state space using a structural vector autoregressive model (Hoover & Demiralp, 2003a; Kilian, 2011). Let $d, T \in \mathbb{N}$ and assume that $X_{i,t}$ is a set of stochastic processes, where $i \in \{1, 2, \dots, d\}$ and $t \in \{1, 2, \dots, T\}$. Let the underlying DAG to be learned be characterized by the set of vertices and edges organised in a pair (V, E), where the vertices are indexed by the set of integers $\{1, 2, \dots, d\}$ and $E \subset V \times V$. Denote the auto-regressive order by $p \in \mathbb{N}$ and let

$$W \in \mathbb{R}^{d,d}, \quad A_i \in \mathbb{R}^{d,d}, \quad i \in \{1, 2, \dots, p\}, \tag{1}$$

be the weighed adjacency matrix of (V, E) and A_i be the matrices encoding the time regressive dependencies. The intra-slice interactions defined at the present time are expressed by the weight matrix W and the inter-slice interactions are expressed by A_i. For the sake of simplicity, the matrices A_i are assumed to be constant. Let $X_t \in \mathbb{R}^{n,d}$ be the data matrix at time t, then the linear auto-regressive model of order p reads

$$X_t = X_t W + X_{t-1} A_1 + X_{t-2} A_2 + \ldots + X_{t-p} A_p + Z, \tag{2}$$

where $Z \in \mathbb{R}^{n,d}$ is the error vector, which is not assumed to be Gaussian. Note that non-linear auto-regressive models can also be formulated in an analogous way. The problem may be written in a simplified manor as

$$X_t = X_t W + Y_t A + Z, \tag{3}$$

where

$$A = A_1 \,|A_2 \ldots |A_p \,, \quad Y_t = X_{t-1} \,|X_{t-2} \ldots |X_{t-p} \tag{4}$$

To maximize the fit of the data over the model, a score function, which reads may be formulated

$$J(W, A) = \|X - XW - YA\|_F^2 + \lambda \|W\| + \eta \|A\|, \tag{5}$$

where $\|\cdot\|$ denotes an arbitrary matrix norm and $\lambda, \eta > 0$ are sufficiently small regularization coefficients. The problem of interest then reads

$$\begin{aligned} &\min_{W,A} J(W, A), \\ &G(W) \in \Gamma_{DAG}, \end{aligned} \tag{6}$$

where A need not be constrained, since cycles are excluded by construction; $\|\cdot\|$ denotes an arbitrary norm, which is usually chosen to be the $l1$-norm and $\|\cdot\|_{\mathbb{F}}$ denotes the Frobenius norm.

Remark 1 *The identifiability of W and A using 6 has been studied for both Gaussian and non-Gaussian noise. Irrespective of the noise, the identifiability of A is a consequence of the basic theory of autoregressive models (Kilian, 2011). The identifiability of W is a bit more involved and must be separated into the Gaussian and non-Gaussian case. In either case, however, identifiability is possible under mild conditions (Hyvärinen et al., 2010; Peters & Bühlmann, 2012).*

3 MIXED INTEGER QUADRATIC PROGRAMMING FORMULATION

Formulating the learning problem as a mixed-integer quadratic problem sets things up so that a globally convergent algorithm may be used. This is fundamental in order for high-precision learning to be possible.

Let $e_{i,j} \in \{0,1\}$ and $e^s_{i,j} \in \{0,1\}$ be decision variables that govern the placement of edges between random variables at time level t and between time levels t and $t-s$, respectively, and let $w_{i,j} \in \mathbb{R}$ and $a^t_{i,j} \in \mathbb{R}$ be the associated edge weights. Using these variables, the scoring function of problem equation 6 becomes

$$J_p = \sum_{i=1}^{n}\sum_{j=1}^{d}\left| X_{i,j} - \sum_{k=1}^{d} X_{i,k} w_{k,j} - \sum_{s=1}^{p}\sum_{k=1}^{d} X^s_{i,k} a^s_{k,j} \right|^2 + \text{REG}, \tag{7}$$

which avoids the use of a bi-linear term if the additional constraints

$$w_{k,j} \leq c e_{k,j}, \quad w_{k,j} \geq -c e_{k,j} \text{ for all } k,j \in \{1,2,\ldots,d\}. \tag{8}$$

and

$$a^s_{k,j} \leq c e^s_{k,j}, \quad a^s_{k,j} \geq -c e^s_{k,j} \text{ for all } k,j \in \{1,2,\ldots,d\}, s \in \{1,2,\ldots,p\} \tag{9}$$

are imposed, where $c > 0$ is the maximal admissible magnitude of any weight and $\lambda > 0$ is a regularization constant.

Where REG is a regularization expression equals either: (L1)

$$\text{REG} = \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} e_{i,j} + \eta \sum_{s=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n} e^s_{i,j}. \tag{10}$$

or (L2)

$$\text{REG} = \lambda \sum_{i=1}^{n}\sum_{j=1}^{n} e_{i,j} + \eta \sum_{s=1}^{p}\sum_{i=1}^{n}\sum_{j=1}^{n} a^s_{i,j}. \tag{11}$$

Lastly, the acyclicity constraints are described. Let C denote the set of all cycles in a graph with d vertices, where each cycle $c \in C$ of length k is represented as a set of edges: $c = \{(i_1,i_2),(i_2,i_3),\ldots,(i_{k-1},i_1)\}$. The constraint excluding a cycle $c \in C$ from a solution then reads

$$\sum_{(i,j)\in c} e_{i,j} \leq k-1. \tag{12}$$

The algorithmic treatment of constraint 12 is key in the following section, in which the algorithmic treatment is discussed as implementing the branch-and-bound-and-cut algorithm without a reduction mechanism for this constraint is doomed to fail due to the super-exponential number of such constraints.

4 ALGORITHMIC IMPLEMENTATION USING BRANCH-AND-BOUND-AND-CUT

One of our main contributions is the development of a branch-and-bound-and-cut algorithm to solve the aforementioned formulation. Since the acyclic constraints 12 needs to only be imposed for the edges of the graph representing the intra-slice level, all of what follows only applied to the intra-slice picture. While we leverage the traditional branch-and-bound approach as described in (Achterberg, 2007, e.g.), we incorporate cycle exclusion constraints equation 12 using "lazy" constraints. These are only enforced once an integer-feasible solution candidate is found. If a violation of a lazy constraint occurs, the constraint is added across all nodes in the branch-and-bound tree. At the root node, only $O(|E|)$ constraints 8 and 9 are initially used. Cycle-exclusion constraints equation 12 are added later. Note that this method is not a heuristic and does not lead to a possibly harmful reduction (or extension) of the solution space leading to omitting possible solutions or returning solutions which are not DAGs. Furthermore, it is shown that the number of constraints that are

actually needed in a computation are many orders of magnitude less than the number of all possible constraints.

Once a new mixed-integer feasible solution candidate is identified, detecting cycles becomes straightforward using depth-first search (DFS). If a cycle is detected, the corresponding lazy constraint equation 12 is added to the problem. The DFS algorithm solves the problem of cycle detection in a worst-case quadratic runtime relative to the number of vertices in the graph, which contrasts with algorithms that separate related inequalities from a continuous relaxation (Borndörfer et al., 2020; Cook et al., 2011), such as the quadratic program in our case. Three variants of adding lazy constraints for the problem were tested.

- Adding lazy constraint only for the first cycle found.
- Adding lazy constraint only for the shortest cycle found.
- Adding multiple lazy constrains for all cycles found in the current iteration in which an integer-feasible solution candidate is available.

The third mentioned variant was found to consistently deliver the best results, despite (Achterberg, 2007, Chapter 8.9). Therefore, it is applied in all of the numerical tests that follow.

5 Data Generation

We generated data in a manner similar to that described in Zheng et al. (2018a) and Pamfil et al. (2020). The evaluation of ExDBN was performed on synthetic data generated as follows. First, a random intra-slice DAG was created using either the Erdős-Rény (ER) model or the scale-free Barabási–Albert (SF) model. The DAG weights were sampled uniformly from the intervals $[-2.0, -0.5] \cup [0.5, 2.0]$.

Next, inter-slice graphs were generated using the ER model. For each inter-slice graph, weights were sampled from the interval $[-0.5\alpha, -0.2\alpha] \cup [0.2\alpha, 0.5\alpha]$, where $\alpha = 1/\eta^{t-1}$, $\eta \geq 1$ is the decay parameter, and t is the time of the slice. $t = 0$ corresponds to the intra-slice, while $t \in \{1, \ldots, p\}$ represents the inter-slices.

Note that we used slightly different generator than in Pamfil et al. (2020) on which DYNOTEARS algorithm perform worse than in original article. We adapted ER and SF generators from Zheng et al. (2018a) for dynamic networks.

6 Numerical Experiments

In recent years, many solvers have been developed to facilitate the graphical learning of Bayesian networks that represent causality (Pamfil et al., 2020; Hyvärinen et al., 2010; Malinsky & Spirtes, 2018; Gao et al., 2022; Dallakyan, 2023; Lorch et al., 2021). Each of these solvers (including the one presented) face the curse of dimensionality, which somewhat restricts the applicability of each solver and thus through testing needs to be provided. It is impossible to test the proposed solution w.r.t. every solver developed. There is, however, a significant branch of development that allows for direct comparison and by transitivity of results the comparison with many previous solvers follows.

In 2020, Pamfil et al. (2020) have developed a locally convergent method, called DYNOTEARS, that learns causality as a Bayesian network that supersedes the solution methods previously developed (Hyvärinen et al., 2010; Malinsky & Spirtes, 2018; Zheng et al., 2018a). Further developments based on previous publications include formulating the problem in the frequency domain or defining differentiable Bayesian structures (Dallakyan, 2023; Lorch et al., 2021). In the following, we provide a head-to-head comparison with DYNOTEARS and thus by transitivity with the methods documented by Hyvärinen et al. (2010); Malinsky & Spirtes (2018).

6.1 Benchmark Setup and Quantities of Interest

In Section 6.2, W_{init} denotes the adjacency matrix representing the intra-slice dependencies and A_{init} denotes the inter-slice dependencies of the ground truth, where A_{init} is used to denote a p-tuple as in

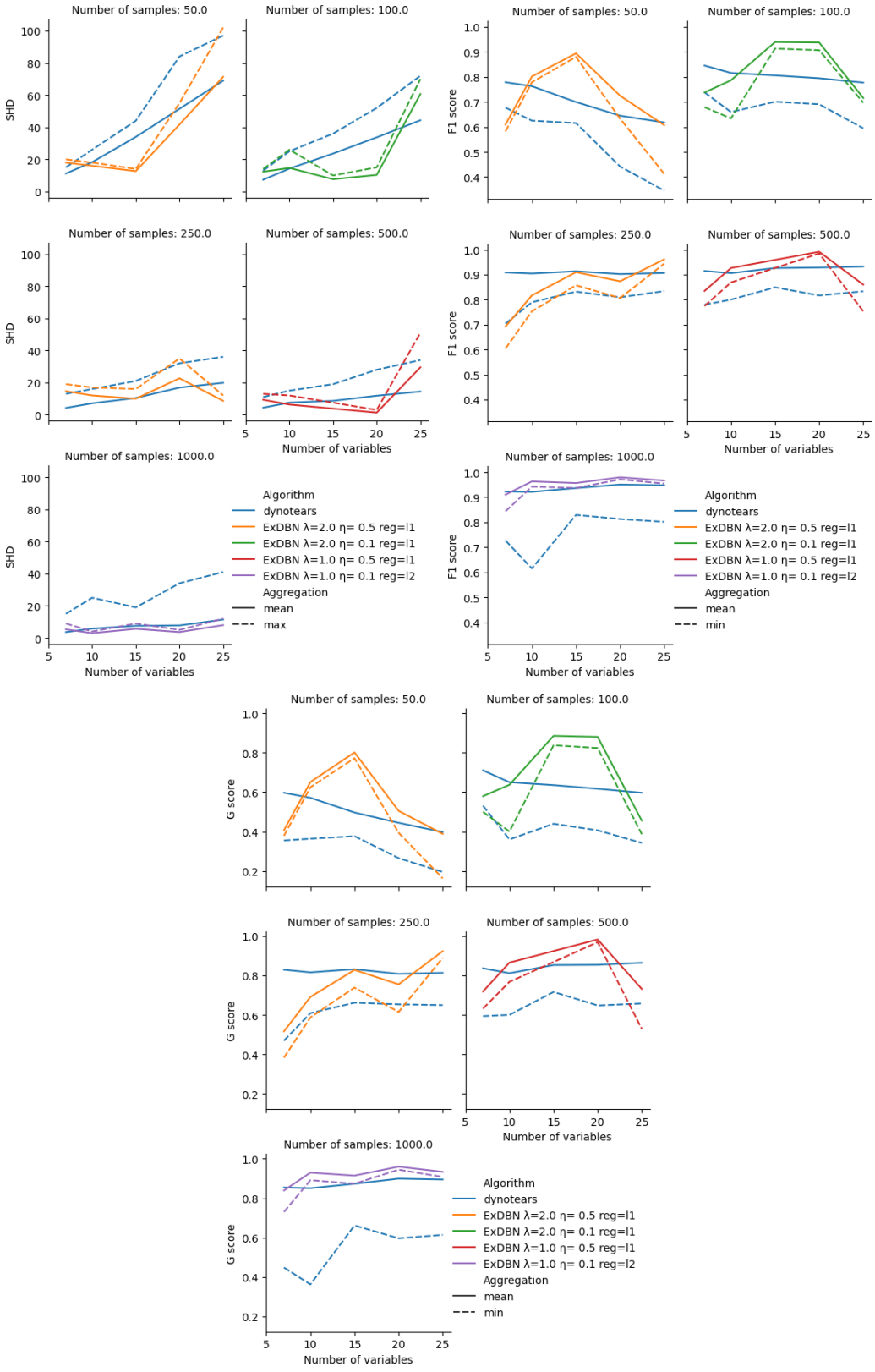

Figure 1: SHD, F1 score, and G score for a test case using ER3-1 random ensemble, i.e., edge-vertex ratio 3 on intra graph, e-v ratio 1 on inter graph, recursion depth 1.

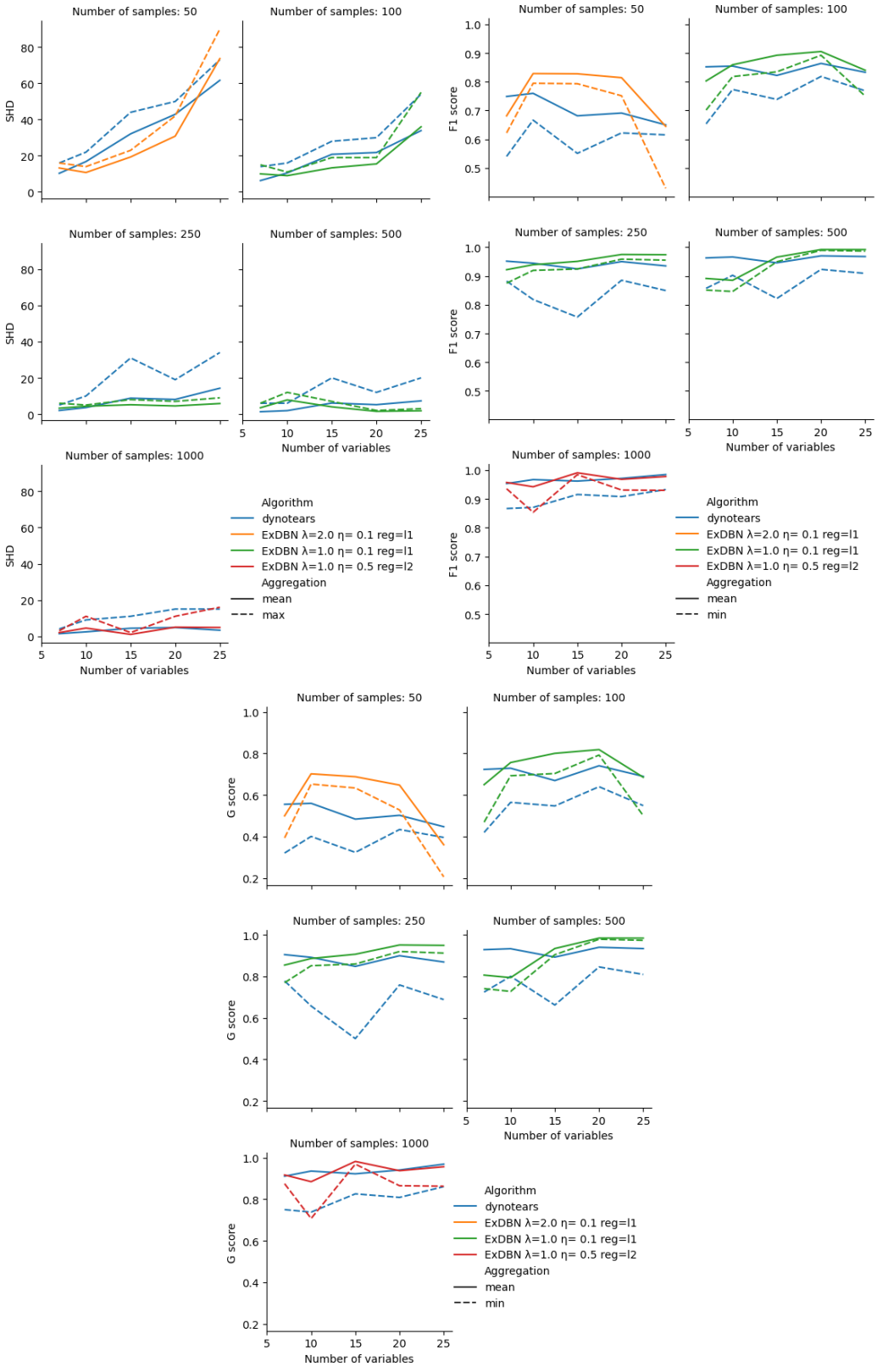

Figure 2: SHD, F1 score, and G score for a test case using SF3-1 random ensemble, i.e., edge-vertex ratio 3 on intra graph, e-v ratio 1 on inter graph, recursion depth 1.

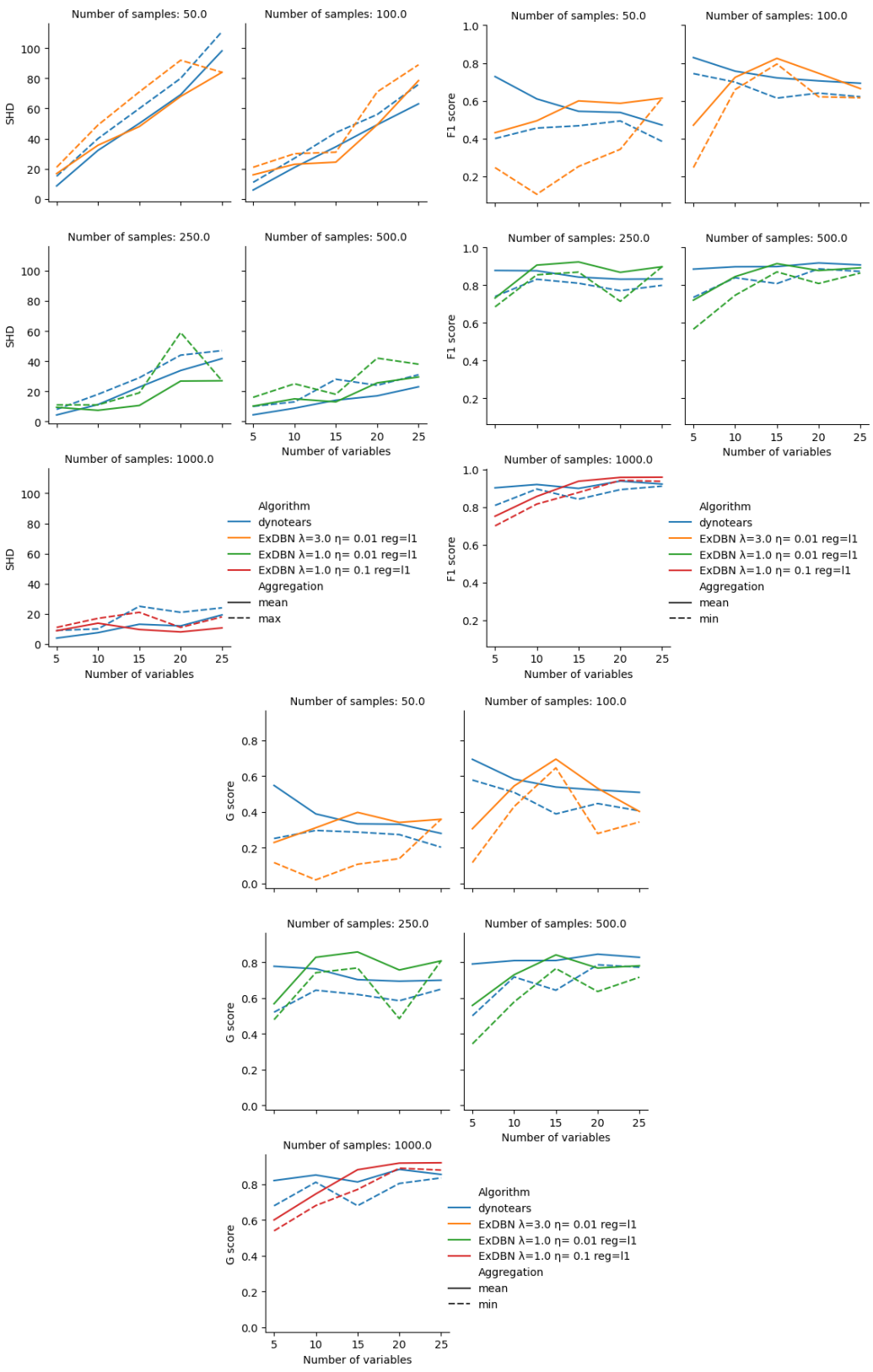

Figure 3: SHD, F1 score, and G score for a test case using ER2-1-1 random ensemble, i.e., edge-vertex ratio 2 on intra graph, e-v ratio 1 on inter graphs, recursion depth 2.

equation 4. W_{init} and A_{init} are then used to generate data while applying noise of Gaussian distribution. Following the data generation process, a matrices W and A are identified and compared with W_{init} and A_{init}. Because the noisy data inevitably leads to some falsely identified edges, typically with negligible weights, one may remove edges with weight less than $\delta > 0$ from the W and A, resulting in a graph W^{δ} and A^{δ}, respectively. To compare methods for known ground truth W_{init} and A_{init}, we choose the best possible $\delta > 0$ for each method. This $\delta > 0$ may then be used as a reference for learning from data, where a ground truth is not known. Next, we introduce the relevant metrics used to evaluate the quality of reconstruction, when W_{init} is available.

In the following, we suppose that a DBN represented by an inter-slice matrix V and inter-slice matrix A is denoted by an ordered pair (V, A). Let (V, A) and (V, A) be two such pairs, then one defines the structural Hamming distance (SHD) as

$$\rho\left(V, A; W, B\right) = \sum_{i,j=1}^{d} r_{ij}\left(V, W\right) + \sum_{k=1}^{p} \sum_{i,j=1}^{d} r_{ij}\left(A_k, B_k\right), \tag{13}$$

where

$$r_{ij}\left(C, D\right) = \begin{cases} 0 & \text{if } C_{ij} \neq 0 \text{ and } D_{ij} \neq 0 \text{ or } C_{ij} = 0 \text{ and } D_{ij} = 0 \\ \frac{1}{2} & \text{if } C_{ij} \neq 0 \text{ and } D_{ji} \neq 0 \\ 1 & \text{otherwise.} \end{cases} \tag{14}$$

SHD is used as a score that describes the structural similarity of two DAGs in terms of edge placement and is commonly used to asses the quality of solutions (Zheng et al., 2018b; Pamfil et al., 2020). Besides SHD

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \text{ and recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}, \tag{15}$$

are used Andrews et al. (2024) to evaluate the quality of structural recovery. It is important to note that precision and recall isolate the false positives and negatives, respectively, opposed to SHD, where these quantities are both accounted for simultaneously. The last metric that can be used to evaluate structural similarity is the F1 score and reads

$$F_1 = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}}. \tag{16}$$

Note that all of the quantities evaluated in equation 15 and equation 16 are a result of summing up all of the differences over both inter and intra slice dependencies between a given pair (V, W) and a grownd truth.

Although structural similarity is a key concern, merely comparing structural properties does not tell the full story, as the weights play a crucial role in the resulting statistical behavior of the found DAG. This motivates us to additionally utilize a cost function based metric, which reads

$$\sigma_p\left(V, W\right) = \left|J_p\left(V\right) - J_p\left(W\right)\right|, \tag{17}$$

where $\lambda = 0$ and typically $p = 2$. We may also evaluate the differences in adjacency matrices by considering

$$\left\|V - W\right\|_{\mathbb{F}}, \tag{18}$$

where $\left\|\cdot\right\|_{\mathbb{F}}$ denotes the Frobenius norm.

6.2 Synthetic Benchmark Results

In the following benchmark, the generation methods described in Section 5 are used to compare ExDBN with DYNOTEARS Pamfil et al. (2020) under the assumption of Gaussian noise. Even though the cost function is a maximum likelihood estimator (see Section 1) for non-Gaussian noise also, we leave this evaluation for a future publication. The scaling is studied for different numbers of variables, samples and graph generation methods with the relevant metrics; SHD, F1 score and G score recorded in Figures 1, 2 and 3. A statistical ensemble with 10 different seeds was used for each of the experiments and the mean and worst possible case value is used in the plots. It should be noted, that naturally the worst possible value and the mean can together be used to bound the variance with

respect. The solution time is capped for ExDBN at 7200 seconds and the regularization applied in ExDBN needs to be scaled appropriately with the number of samples as it is assumed that the optimal choice of regularization constant is a decreasing function of the number of samples. We use the aforementioned as a guide (in a non-strict way) to find the right regularization for a given sample size. This follows from the fact that the regularization is to be kept proportionally small to the main objective expressed by equation 7. Furthermore, it was found that changing the regularization from L1 to L2 is beneficial for identification when the number of samples is large. Furthermore, if we don't know the ground truth graph. We can try to run the algorithm for multiple values of λ and η and then use the one which yields better MIP GAP. For smaller number of samples the regularization L1 works better. For bigger number of samples L2 yields good results and it is usually faster.

The tests results may be divided into two categories by the average number of edges. Figures 1 and 2 show higher degree graphs (average degree 3) and Figure 3 depicts the reconstruction of a lower degree graph. In the case of the lower degree graph, it is clear that both DYNOTEARS and ExDBN perform similarly with ExDAG beating out DYNOTEARS some of the time with the converse being true equally often. In the case of the identification of higher degree graphs, however, one can notice that the worst possible performance and the mean performance are much closer in the case of ExDBN, where we can point out for instance the G score in the case with 1000 samples. In these instances the differences between the worst possible G score difference is between 0.3 and 0.5 in the case of DYNOTEARS but stays well under 0.1 in the case of ExDBN. The aforementioned can be interpreted as superior reliability of solution as the worst possible reconstruction is consistently better. Focusing on the 1000 sample case still, while somewhat taking into account the previous ones also, we see that the performance gap between the solvers increases in favor of ExDBN as we increase the number of samples. In the lower sample cases, one may also observe that ExDBN outperforms DYNOTEARS for many graph sizes in the mean and consistently outperforms DYNOTEARS in the worst possible case (min/max depending on metric). Note that the global convergence of the method, which is rooted in the fundamentals of mixed integer quadratic programming, allows us to increase computation time, which leads to improving the metrics reported further. While some time-sensitive applications like short term stock evaluation might not be able to benefit from this, others like biomedical applications might benefit as a computation lasting several days, in which the accuracy in measurably improved (by monitoring the duality gap), is desirable.

6.3 Application in Biomedical Sciences

In biomedical sciences, there is a keen interest in learning dynamic Bayesian networks with the view of estimating causal effects (Tennant et al., 2020) and identifying confounding variables that require conditioning. A recent metaanalysis (Tennant et al., 2020) of 234 articles on learning DAGs in biomedical sciences found that the averaged DAG had 12 nodes (range: 3–28) and 29 arcs (range: 3–99). Interestingly, none of the DAGs were as sparse as the commonly considered random ensembles; median saturation was 46%, i.e., each of all possible arcs appeared with probability 46% and does not converge to a global minimum of the problem.

As an example, we consider a recently proposed benchmark of Ryšavý et al. (2024), where Krebs cycle is to be reconstructed from time series of reactant concentrations of varying lengths. There, DYNOTEARS cannot reach (Ryšavý et al., 2024) F1 score of 0.5 even with very long time series. In contrast, our method can solve instances equation 7 to global optimality. Using ExDBN, however, the global minimization is ensured given sufficient time and thus the maximum likelihood estimator is found. It should be noted, however, that depending on the number of samples and noise, it may be that even the maximum likelihood estimator is not sufficiently accurate. This does not however reflect poorly on the method itself, but it rather a matter of the modification of data collection methods associated with the experiment. In one hour time limit, ExDBN can find a solution with 38% duality gap.

6.4 Application in Finance

In financial services, there is also a number of important applications. The original DYNOTEARS paper considered a model of diversifying investments in stocks based on dynamic Bayesian networks. Independently, Ballester et al. (2023) consider systemic credit risk, which is one of the most important concerns within the financial system, using dynamic Bayesian networks. They found that

the transport and manufacturing sectors transmit risk to many other sectors, while the energy sector and banking receive risk from most other sectors. To a lesser extent, there is risk transmission present between approximately 25% of the pairs of sectors, and these network relationships explain between 5 % and 40 % of single systemic risks. Notice that these instances are much denser than the commonly used random ensembles.

We elaborate upon the example of Ballester et al. (2023), where 10 time-series capture the spreads of 10 European credit default swaps (CDS). Considering the strict licensing terms of Refinitiv, the data from Ballester et al. (2023) are not available from the authors, but we have downloaded 16 time-series capturing the spreads of 16 European CDS with RED6 codes 05ABBF, 06DABK, 0H99B7, 2H6677, 2H66B7, 48DGFE, 6A516F, 8A87AG, 8B69AP, 8D8575, DD359M, EFAGG9, FF667M, FH49GG, GG6EBT, NN2A8G, from January 1st, 2007, to September 25th, 2024. This amounts to over 11 MB of time-series data, when stored as comma-delimited values in plain text. While the procedure for learning the dynamic Bayesian network in Ballester et al. (2023) is rather heuristic, we can solve the mixed-integer programming (MIP) instance for the 16 European CDS within 5 minutes. In the heuristic of Ballester et al. (2023), they first perform unconditional independence tests on each set of two time series containing an original series and a lagged time series, to prune the subsequent number of unconditional independence tests performed. There are 45 unconditional and conditional independence tests performed first, to suggest further 200 conditional independence tests. We stress that the procedure of Ballester et al. (2023) does not come with any guarantees, while our instance equation 7 is solved to global optimality. Furthermore, on the systemic-risk transmission instance above, we are able to arrive at a global optimum within 2 minutes (using L2 regularization), which validates the scalability of mixed-integer programming solvers.

7 Conclusion

Dynamic Bayesian networks have wide-ranging applications, including those in biomedical sciences and computational finance, illustrated above. Unfortunately, their use has been somewhat limited by the lack of well-performing methods for learning those. Our method, ExDBN, provides the best possible estimate of the DBN, in the sense of minimizing empirical risk equation 7. Significantly, our method does not suffer much from the curse of dimensionality even for real-world, dense instances, which are typically challenging for other solvers. This is demonstrated most clearly in the systemic-risk transmission use-case detailed in Section 6.4, in which the global minimizer is found within 2 minutes. Additionally, the use of the guarantees on the distance to the global minimizer (so-called MIP gap, available ahead of the convergence to the global minimizer) provides a significant tool for fine-tuning the parameters of the solver in the case of real-world application, where the ground truth is not available. Combined with global convergence guarantees of the maximum likelihood estimator, this provides a robust method, with state-of-the-art statistical performance.

References

Tobias Achterberg. *Constraint Integer Programming*. PhD thesis, Technische Universitaet Berlin, 2007.

Bryan Andrews, Joseph Ramsey, Ruben sanchez romero, Jazmin Camchong, and Erich Kummerfeld. Fast scalable and accurate discovery of dags using the best order score search and grow-shrink trees. *Advances in neural information processing systems*, 36:63945–63956, 05 2024.

Charles Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 02 2022. doi: 10.1613/jair.1.13428.

Laura Ballester, Jesúa López, and Jose M. Pavía. European systemic credit risk transmission using bayesian networks. *Research in International Business and Finance*, 65:101914, 2023. ISSN 0275-5319. doi: https://doi.org/10.1016/j.ribaf.2023.101914. URL https://www.sciencedirect.com/science/article/pii/S0275531923000405.

Alexis Bellot, Kim Branson, and Mihaela van der Schaar. Neural graphical modelling in continuous-time: consistency guarantees and algorithms. In *International Conference on Learning Representations*, 2021. URL https://api.semanticscholar.org/CorpusID:246485884.

Ralf Borndörfer, Heide Hoppmann, Marika Karbstein, and Niels Lindner. Separation of cycle inequalities in periodic timetabling. *Discrete Optimization*, 35:100552, 2020.

William J Cook, David L Applegate, Robert E Bixby, and Vasek Chvátal. *The traveling salesman problem: a computational study*. Princeton university press, 2011.

Aramayis Dallakyan. On learning time series summary dags: A frequency domain approach. 2023. URL https://api.semanticscholar.org/CorpusID:258179448.

Thomas L. Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5, 1989. URL https://api.semanticscholar.org/CorpusID:57798167.

Selva Demiralp and Kevin Hoover. Searching for the causal structure of a vector autoregression*. *Oxford Bulletin of Economics and Statistics*, 65:745 – 767, 12 2003. doi: 10.1046/j.0305-9049.2003.00087.x.

Tian Gao, Debarun Bhattacharjya, Elliot Nelson, Miaoyuan Liu, and Yue Yu. Idyno: Learnmalinskying nonparametric dags from interventional dynamic data. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:250340690.

Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods. *ACM Computing Surveys*, 53(4):1–37, July 2020. ISSN 1557-7341. doi: 10.1145/3397269. URL http://dx.doi.org/10.1145/3397269.

Kevin Hoover and Selva Demiralp. Searching for the causal structure of a vector autoregression. *SSRN Electronic Journal*, 04 2003a. doi: 10.2139/ssrn.388840.

Kevin D. Hoover and Selva Demiralp. Searching for the causal structure of a vector autoregression. *Macroeconomics eJournal*, 2003b. URL https://api.semanticscholar.org/CorpusID:16111786.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11:1709–1731, 07 2010.

Lutz Kilian. Structural Vector Autoregressions. CEPR Discussion Papers 8515, C.E.P.R. Discussion Papers, August 2011. URL https://ideas.repec.org/p/cpr/ceprdp/8515.html.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. 01 2009. ISBN 978-0-262-01319-2.

Lars Lorch, Jonas Rothfuss, Bernhard Scholkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *ArXiv*, abs/2105.11839, 2021. URL https://api.semanticscholar.org/CorpusID:235187432.

Helmut Luetkepohl. *The New Introduction to Multiple Time Series Analysis*. 01 2005. ISBN 978-3-540-40172-8. doi: 10.1007/978-3-540-27752-1.

Daniel Malinsky and Peter Spirtes. Causal structure learning from multivariate time series in settings with unmeasured confounding. In Thuc Duy Le, Kun Zhang, Emre Kıcıman, Aapo Hyvärinen, and Lin Liu (eds.), *Proceedings of 2018 ACM SIGKDD Workshop on Causal Disocvery*, volume 92 of *Proceedings of Machine Learning Research*, pp. 23–47. PMLR, 20 Aug 2018. URL https://proceedings.mlr.press/v92/malinsky18a.html.

Tom Michoel and Jitao David Zhang. Causal inference in drug discovery and development. *Drug Discovery Today*, 28(10):103737, 2023. ISSN 1359-6446. doi: https://doi.org/10.1016/j.drudis.2023.103737. URL https://www.sciencedirect.com/science/article/pii/S1359644623002532.

Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, 01 2002.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Paul Beaumont, Konstantinos Georgatzis, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, 2020. URL https://api.semanticscholar.org/CorpusID:211010514.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101, 05 2012. doi: 10.1093/biomet/ast043.

Jagath Rajapakse and Juan Zhou. Learning effective brain connectivity with dynamic bayesian networks. *NeuroImage*, 37:749–60, 10 2007. doi: 10.1016/j.neuroimage.2007.06.003.

Petr Ryšavý, Xiaoyu He, and Jakub Mareček. Causal learning in biomedical applications: A benchmark, 2024. URL https://arxiv.org/abs/2406.15189.

Xiangyu Sun, Guiliang Liu, Pascal Poupart, and Oliver Schulte. Nts-notears: Learning nonparametric temporal dags with time-series data and prior knowledge. *ArXiv*, abs/2109.04286, 2021. URL https://api.semanticscholar.org/CorpusID:237454655.

Peter W G Tennant, Eleanor J Murray, Kellyn F Arnold, Laurie Berrie, Matthew P Fox, Sarah C Gadd, Wendy J Harrison, Claire Keeble, Lynsie R Ranker, Johannes Textor, Georgia D Tomova, Mark S Gilthorpe, and George T H Ellison. Use of directed acyclic graphs (DAGs) to identify confounders in applied health research: review and recommendations. *International Journal of Epidemiology*, 50(2):620–632, 12 2020. ISSN 0300-5771. doi: 10.1093/ije/dyaa213. URL https://doi.org/10.1093/ije/dyaa213.

Marcel A.J. van Gerven, Babs G. Taal, and Peter J.F. Lucas. Dynamic bayesian networks as prognostic models for clinical patient management. *Journal of Biomedical Informatics*, 41(4):515–529, 2008. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2008.01.006. URL https://www.sciencedirect.com/science/article/pii/S1532046408000154.

Alessandro Zandonà, Rosario Vasta, Adriano Chió, and Barbara Di Camillo. A dynamic bayesian network model for the simulation of amyotrophic lateral sclerosis progression. *BMC Bioinformatics*, 20, 04 2019. doi: 10.1186/s12859-019-2692-x.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018a. URL https://arxiv.org/abs/1803.01422.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018b.

Q. Zhong, Y. Cheng, Z. Li, D. Wang, C. Rao, Y. Jiang, L. Li, Z. Wang, P. Liu, Y. Zhao, P. Li, J. Suo, Q. Dai, and K. He. Ultra-efficient causal learning for dynamic csa-aki detection using minimal variables. In *medRxiv*, 2023. URL https://api.semanticscholar.org/CorpusID:265657431.