Using Score-based Structure Learning to Computationally Learn Direct Influence between Hierarchical Dynamic Bayesian Networks

Ritesh Ajoodha
School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg
South Africa
Email: ritesh.ajoodha@wits.ac.za

Benjamin Rosman
School of Computer Science and Applied Mathematics
The University of the Witwatersrand, Johannesburg
South Africa
Email: benjamin.rosman@wits.ac.za

***Abstract*—Numerous fields of science have investigated stochastic processes which are partially observable. However, the discovery and analysis of the interaction between, and the influence upon each other, of several of these processes, have not been probed extensively. This paper uses probabilistic structure learning in an attempt to learn influence relationships between stochastic processes that are partially observed. These processes are represented by hierarchical dynamic Bayesian networks (H-DBNs). To track the direct influence between the these processes, we provide an algorithm that extends the BIC structure score as well as the cumbersome (greedy hill-climbing) local search procedure. Our method leverages the temporal nature of the HDBN through the use of *assembles* thereby surpassing the standard approach that treats each process as a single variable. The derived BIC-score for HDBN families is clearly shown to be theoretically *decomposable* and empirically *consistent*.**

***Index Terms*—Learning influence networks, Structure learning, Hidden Markov models, Stochastic processes.**

I. Introduction

Stochastic processes are often used for narrating the progression of variables with respect to time. In spite of this, the reciprocal influences between several of these processes is not addressed much in the literature. *Constraint-based* structure learning can view the influence relation between processes as a collection of independence assumptions. It is assumed, from this constraint-based approach, that a perfect map can be obtained from this collection.

This can be achieved using statistical tests for conditional independence [1]. Applications that use constraint-based methods that try to learn the structure between processes include: discovering micro-behaviour in econometric models [2], identifying interacting networks in the brain [3], and for causal interpretation between statistical models [4].

Statistical tests frequently are incorrect with regard to determined independence assertions. This oversights, while performing multiple hypothesis testing, can spread into the structure of the network which subsequently decreases the likelihood of recovering the true probabilistic graphical influence network [5]. This paper proposes an alternate approach as being better suited to this task since it:

1) looks at all the possible influence relations between processes as a single state in the search space;
2) maintains essential score properties which allows for feasible computations;
3) and it presents an unambiguous signal about the independence assertions between dynamic models with respect to the data.

More formally, we can define the concept of direct influence between two processes, $A_{\rightarrow}$ and $B_{\rightarrow}$, as a *proportional* relationship between them. More specifically, this *proportional* relationship suggests that changing some value in some point in time t_1 in $A_{\rightarrow}$ will 'directly' impact the concurrent value at the same point in the time t_1 in $B_{\rightarrow}$.

This 'direct' influence is defined with respect to *delayed influence* where changes in a value in process $A_{\rightarrow}$ at time t_1 will only result in a change in the subsequent value at or after t_1 in $B_{\rightarrow}$.

This paper provides the very first score-based structure learning modelling of this problem to find the complete *direct influence network* (DIN), which is a dynamic Bayesian network (DBN) which factorises over a joint distribution between a finite collection of stochastic and partially observable processes represented as hierarchical dynamic Bayesian networks (H-DBNs).

The following two assumptions are made in our search for the optimal DIN:

- Firstly, we will assume that the data, $\mathcal{D} = \{\xi_1, \ldots, \xi_M\}$, is sampled IID (over time) from some underlying dynamic distribution $P^*(\mathcal{H})$, where $\mathcal{H} = \{\mathcal{H}_1(\mathcal{X}), \ldots, \mathcal{H}_K(\mathcal{X})\}$ is a collection of H-DBNs over the collection of random variables $\mathcal{X} = \{X_1, \ldots, X_N\}$.
- Secondly, we assume that $P^*(\mathcal{H})$ is produced by another DBN, $\mathcal{G}^*(\mathcal{H})$, which is called the *ground truth* structure.

This paper aims to recover the local independence assertions in $\mathcal{G}^*(\mathcal{H})$, denoted $\mathcal{I}_\ell(\mathcal{G}^*(\mathcal{H}))$, by only observing $\mathcal{D}$. The

978-1-5386-5541-2/21/$31.00 ©2021 IEEE

importance of learning a DBN Bayesian network structure is determined by its motivation for use:

- Attempts to learn the ground truth DIN structure (knowledge discovery) means precisely stating $\mathcal{I}_\ell(\mathcal{G}^*(\mathcal{H}))$, then one should accept that there are many *perfect maps* for $P^*(\mathcal{H})$ that exist in $\mathcal{D}$ [6]. It is well accepted that identifying $\mathcal{I}_\ell(\mathcal{G}^*(\mathcal{H}))$ from $\mathcal{G}^*(\mathcal{H})$'s collection of Bayesian networks, which will provide identical "fit" to data, is *not identifiable* from the data-set $\mathcal{D}$ given that every I-equivalent structure will yield an identical likelihood for $\mathcal{D}$. This means at best one may wish to recover the I-equivalence class of $\mathcal{G}^*(\mathcal{H})$. However, this is difficult since data sampled from $P^*(\mathcal{H})$ will never perfectly recreate the independence assumptions of $\mathcal{G}^*(\mathcal{H})$.
- If we are searching for a DIN for *density estimation*, which is to approximate a model which is similar to the underlying distribution $P^*(\mathcal{H})$ in order to answer probabilistic queries (such as calculating a conditional probability given some values as evidence). In this case we need to be wary of two possibilities:
  - if one specifies more independence assumptions (between processes) than those already captured in $\mathcal{I}_\ell(\mathcal{G}^*(\mathcal{H}))$, we may still be able to capture $P^*(\mathcal{H})$ using some arrangement of the model parameters. Conversely, a specification of more independence assumptions than $\mathcal{I}_\ell(\mathcal{G}^*(\mathcal{H}))$, may result in *data fragmentation*.
  - Alternatively, selecting few edges may result in restricting the model to never being able to capture the true empirical distribution $P^*(\mathcal{H})$. However, few edges implies a sparser structure than one with more edges which avoids fragmentation.

  In practice, often less edges are chosen for density estimation given that better generalisation is gained for new instances [5].

The significance of this study are broad. DINs for partially observable processes can express complex relationships and reveal how processes effect each other.

For example, DINs can represent how traffic in road networks manifest in data. In educational data mining we can reveal the influence between participants in a lecture venue using a DIN. Density estimate can reveal the implications of students success based on modelled these interactions. Alternatively, we can model influence between end-users in an IoT network [**?**], [7]; or perhaps learn the influence between learned skills in a human or robot [**?**].

The method in this paper expands notions in score-based structure learning for tracking direct influence between partially observable processes by:

1) factorising the collection of processes into a collection of H-DBNs;
2) and thereafter, defines an *assemble* relation between proceeses and a *scoring function* to evaluate the quality of candidate DINs.

After these two steps, we consider the well defined combinatorial optimisation problem: to search through the search space for the DIN which optimises the score – which we return as the goal structure with respect to $\mathcal{D}$.

This paper makes the following technical contributions:

1) to the best of our knowledge, this paper provides the first score-based algorithm to learn a *DIN structure* between a collection of processes;
2) we expand one the traditional BIC score to one which scores H-DBNs;
3) we show that the new BIC score for H-DBNs is theoretically decomposable and empirically consistent;
4) we expand on the traditional greedy heuristic search procedure to one which uses assembles to link H-DBNs meaningfully while preserving decomposability and score-equivalence necessary for a feasible search.

The following structure is followed by this paper: Section II presents how each process is learned (Section II-A), the assemble for direct influence (Section II-B), the derived BIC score for H-DBNs (Section II-C), and the structure search procedure (Section II-D); and then, Section III and Section IV review the results and conclusion of this work respectively.

II. The Structure Selection Algorithm

Traditional *Score-based* structure learning involves establishing a conjecture space of candidate networks; establishing a scoring metric which calculates the network-to-observed data compatibility; and an algorithm to distinguish networks to optimise the score as a well-defined optimisation problem.

However, since the search space is super-exponential in size, this poses an NP-hard problem which can be partially solved using heuristic procedures. This section outlines the score-based structure learning algorithm employed in this study which is used to construct the DINs between the collection of input processes.

A. Hierarchical Dynamic Bayesian Networks (H-DBNs)

Stochastic processes are random variables with statistical dependencies between them that unfold over time. The complex probability density can be modelled using dynamic Bayesian networks (DBNs).

This paper extends the traditional DBN into a H-DBNs which is able to encode the associations between the random variables using the language of probabilistic graphical modelling. The H-DBNs can be defined as a two-time-slice hierarchical Bayesian network:

Definition 1. *A hierarchical Bayesian network (HBN) is pair $\mathcal{H} = (\mathcal{G}_X, P_{\mathcal{G}_X})$ where $P_\mathcal{G}$ is a distribution that factorizes over the hierarchical structure $\mathcal{G}_\mathcal{X}$.*

Definition 2. *A two-time-slice hierarchical Bayesian network (2-THBN) for a process over $\mathcal{X}$ is a HBN over $\mathcal{X}'$ given $\mathcal{X}_I$, where $\mathcal{X}_I \subseteq \mathcal{X}$ is a collection of interface variables and $\mathcal{X}'$ is the next time-slice.*

Definition 3 (Hierarchical dynamic Bayesian networks)**.** *A hierarchical dynamic Bayesian network (H-DBN) is a tuple*

$\mathcal{H}_{DB} = \langle \mathcal{H}_0, \mathcal{H}_{\rightarrow} \rangle$, where $\mathcal{H}_0$ is a HBN over $\mathcal{X}^{(0)}$, representing the initial distribution over states and $\mathcal{H}_{\rightarrow}$ is a 2-THBN for the process. Given a time interval $T \geq 0$, the distribution over $\mathcal{X}^{(0:T)}$ is described as a unrolled HBN, where for any $i = 1, \ldots, n$: the Bayesian structure and conditional probability distributions (CPDs) of $\mathcal{X}_i^{(0)}$ are identical for $\mathcal{X}_i$ in $\mathcal{H}_0$; and the Bayesian structure and CPDs of $\mathcal{X}_i^{(t)}$ for $t > 0$ are identical for the next time-slice $\mathcal{X}_i^{'}$ in $\mathcal{H}_{\rightarrow}$.

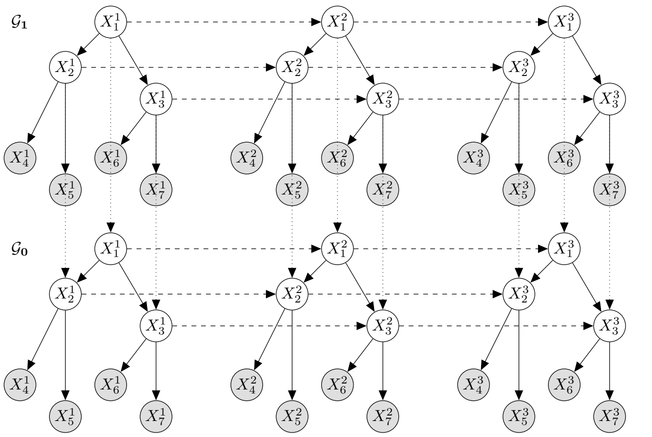

Figure 1: A direct influence assemble between two H-DBNs, $\mathcal{G}_1 \rightarrow \mathcal{G}_0$, with three time-slices.

B. The Direct Assemble

We have so far discussed a common representation for a temporal process using the notion of a HDBN. We now define the assemble relation which describes direct influence between a family of H-DBNs.

We introduce the *direct assemble* which imposes an added collection of independence assertions between two H-DBNs, $\mathcal{H}_1$ and $\mathcal{H}_2$, by inserting an edge which is directed from all the latent (not observed) variables in $\mathcal{H}_1$ to all the *corresponding* latent variables in $\mathcal{H}_2$, given that $\mathcal{I}_\ell(\mathcal{H}_1)$ is the same as $\mathcal{I}_\ell(\mathcal{H}_2)$, Figure 1 illustrates this relation. The notation used in Figure 1 are as follows:

- the intra-time-slices are shown as solid lines;
- the persistent inter-time-slice edges are shown using broken lines;
- lastly, the assemble are provided by the dotted-lined edges.

We formally define this assemble as:

Definition 4 (The direct assemble)**.** *Consider two HBNs, $\mathcal{H}_0 = (\mathcal{G}^1, P_{\mathcal{H}_0})$ and $\mathcal{H}_1 = (\mathcal{G}^2, P_{\mathcal{H}_1})$, where $\mathcal{G}^0$ is the hierarchical network structure whose nodes represent random variables $O_1^{\mathcal{G}^0}, \ldots, O_k^{\mathcal{G}^0}, L_{k+1}^{\mathcal{G}^0}, \ldots, L_m^{\mathcal{G}^0}$, and $\mathcal{G}^1$ is a hierarchical network structure whose nodes represent random variables $O_1^{\mathcal{G}^1}, \ldots, O_k^{\mathcal{G}^1}, L_{k+1}^{\mathcal{G}^1}, \ldots, L_m^{\mathcal{G}^1}$, where both $\mathcal{G}^0$ and $\mathcal{G}^1$ encode the same collection of independence assumptions, and O and L represent the observable and latent variables respectively. Then the assemble structure of $\mathcal{H}_0$ to $\mathcal{H}_1$, denoted $\mathcal{H}_{0\rightarrow 1}$, is defined as $\mathcal{H}_{0\rightarrow 1} = (\mathcal{G}_{0\rightarrow 1}, P_{\mathcal{H}_{0\rightarrow 1}})$, where $\mathcal{G}^{0\rightarrow 1}$ is a hierarchical network structure whose nodes represent the random variables $O_1^{\mathcal{G}^{0\rightarrow 1}}, \ldots, O_i^{\mathcal{G}^{0\rightarrow 1}}, L_{i+1}^{\mathcal{G}^{0\rightarrow 1}}, \ldots, L_n^{\mathcal{G}^{0\rightarrow 1}}$ with the additional independence assumptions: $\forall$ Latent variables L_i: ($L_i^{\mathcal{G}^1} \perp\!\!\!\perp$ NonDescendants$_{L_i^{\mathcal{G}^1}} | L_i^{\mathcal{G}^0}, Pa_{\mathcal{H}_0}^{\mathcal{G}^1}$). In other words each variable L_i is conditionally independent to its nondescendants given its parents.*

The latent (not observed) variables need to be learned when a structure is proposed by the assembled configuration. This assembled configuration expounds direct influence in a straightforward implementation between models, and provides a sparse representation between families of H-DBNs.

More importantly, it provides an instinctive way of expressing the way ‘direct’ influence is expected to flow from one time slice to another, where the choice of granularity proposes a trade-off between generalised and circumstantial direct influence structures.

Algorithm 1: The Direct Influence Assemble

```
procedure FamilyScore($\mathcal{H}_{DB}$ $\mathcal{H}_0, \mathcal{H}_{DB}[]$ $\mathcal{H}_d$)
    score = 0
    if !isEmpty($\mathcal{H}_d$) then
        for each time-slice, $t$, in $\mathcal{H}_0$ do
            for each variable, $x$, in $\mathcal{H}_0[t]$ do
                if isLatent($\mathcal{H}_0[t][x]$) then
                    $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]} = \{\}$
                    if hasInterDep($\mathcal{H}_0[t][x]$) then
                        $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]}$ = {InterDep($\mathcal{H}_0[t][x]$),
                                ExtDep($\mathcal{H}_0[t][x]$)}
                    else
                        $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]} = \{$ExtDep($\mathcal{H}_0[t][x]$)$\}$
                    end if
                    score += $\mathbf{S}_{\hat{P}}(\mathcal{H}_0[t][x], \mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]})$
                else if isObs($\mathcal{H}_0[t][x]$) then
                    if hasInterDep($\mathcal{H}_0[t][x]$) then
                        $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]} = \{$InterDep($\mathcal{H}_0[t][x]$)$\}$
                        score += $\mathbf{S}_{\hat{P}}(\mathcal{H}_0[t][x], \mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]})$
                    else
                        score += $\mathbf{S}_{\hat{P}}(\mathcal{H}_0[t][x], \{\})$
                    end if
                end if
            end for
        end for
    else
        for each time-slice, $t$, in $\mathcal{H}_0$ do
            for each variable, $x$, in $\mathcal{H}_0[t]$ do
                if hasInterDep($\mathcal{H}_0[t][x]$) then
                    $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]} = \{$InterDep($\mathcal{H}_0[t][x]$)$\}$
                    score += $\mathbf{S}_{\hat{P}}(\mathcal{H}_0[t][x], \mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]})$
                else
                    score += $\mathbf{S}_{\hat{P}}(\mathcal{H}_0[t][x], \{\})$
                end if
            end for
        end for
    end if
```

```
37: end procedure
```

A decomposable score for a complete network is one which can also be written as a sum of family scores for the complete network. Algorithm 1 provides a procedure to compute the family score of a collection of H-DBNs given the direct influence assemble using any decomposable score, where $\mathbf{U}^{\mathcal{G}}_{\mathcal{H}_0[t][x]}$ is a collection of dependency variables for variable x in time-slice t in HDBN $\mathcal{H}_0$ and $\mathbf{S}_{\hat{P}}$ is the score produced given the empirical distribution.

We see that on line 9, variables with inter-time-slice dependencies have parents from previous time-slices and from external dependency models. However, as seen on line 15, observable variables do not have inter-times-slice dependencies since they are relatively instantaneous compared to our time granularity and so only have intra-time-slice dependencies to latent variables at higher hierarchical positions.

C. Bayesian Information Criterion

This paper attempts to recover a direct influence network (DIN) between Bayesian temporal models as a well defined optimisation problem: a score is established which measures potential DIN structures relative to the data-set $\mathcal{D}$. We then use heuristic search procedures to find the highest scoring DIN structure.

Many attempts have been made to design scores for this learning task, including the likelihood score and the Bayesian information criterion (BIC) score [8]. Here we extend the traditional likelihood score to one which evaluates a DIN relative to $\mathcal{D}$ by using an assemble relation instead of a standard Bayesian network. We further show that the derived score is decomposed for these DINs.

The likelihood score, sometimes referred to as the log-likelihood score, computes the 'averaged distance' between the empirical joint distribution, $\hat{P}(x_i^{\mathcal{H}_k^{(t)}}, \mathbf{u}_i^{\mathcal{H}_k^{(t)}})$, relative to the product of marginals, $\hat{P}(\mathbf{u}_i^{\mathcal{H}_k^{(t)}})\hat{P}(x_i^{\mathcal{H}_k^{(t)}})$, which relates the collection $\mathcal{I}_\ell(\mathcal{G})$ to $\mathcal{D}$. In the rare case that the two models $\mathcal{H}_0$ and $\mathcal{H}_1$ are independent in $\mathcal{D}$, which almost never happens in empirical data, the proposed score never prefers the simpler network over the more complicated one, since $\text{score}_l(\mathcal{G}_{\mathcal{H}_0 \rightarrow \mathcal{H}_1} : \mathcal{D}) \geq \text{score}_l(\mathcal{G}_\emptyset : \mathcal{D})$. Therefore, the likelihood score can not generalise to the data samples from the empirical ground truth distribution, $P^*(\mathcal{H})$, presenting an over-fitting problem.

To circumvent this, the BIC is often used as a replacement for the likelihood score [8]. The BIC score is a variation of the likelihood score with a which prefers simpler structures, however, it is willing to consider more complex structures only if there is sufficient justification in the data [5].

In other words, the BIC score is willing to mathematically trade-off the fit to data for model complexity - and vice-versa depending on how much data is seen by the model - in doing so decrease over-fitting. In this research the BIC score is adapted to measure the trade-off between the fit to data of a DIN with its complexity. The following score is proposed:

$$\begin{aligned}\text{score}_{BIC}(\mathcal{H}_0 : \mathcal{D}) =& M \sum_{k=1}^{K} (\sum_{t=1}^{T} (\sum_{i=1}^{N} (\mathbf{I}_{\hat{P}}(X_i^{\mathcal{H}_k^{(t)} \rightarrow \mathcal{H}_0^{(t)}}; \\ & \mathbf{U}^{\mathcal{H}_0}_{X_i^{\mathcal{H}_k^{(t)} \rightarrow \mathcal{H}_0^{(t)}}})))) \\ & - \frac{\log M}{c} DIM[\mathcal{H}_0],\end{aligned} \tag{1}$$

where M is the size of the data-set; K is the size of the dependency model set; T is count of time-slices for each dependency model; N is the size of the variable set in each time-slice; $\mathbf{I}_{\hat{P}}$ is the measure of mutual dependence (mutual information) with regard to the empirical distribution; $DIM[\mathcal{H}_0]$ is the size of the set of independent parameters in $\mathcal{H}_0$; and $\rightarrow$ is the symbol for the assemble relation.

Equation 1 is derived in Preposition 1. Preposition 1 reveals that Equation 1 is nothing more than the traditional likelihood score (fit to data) with an added penalty term. In Preposition 1:

1) The entropy term, $\mathbf{H}_{\hat{P}}$, is independent from the structure chosen and is therefore negligible.
2) The result trades-off the fit to data with the complexity of the DIN.

In Preposition 1, there are two notable behaviours regarding the growth rates of the terms considered:

1) **Mutual information:** The term $\mathbf{I}_{\hat{P}}$ grows linearly with respect to the number of samples considered in $\mathcal{D}$.
2) **Complexity:** The term $\frac{\log M}{c} DIM[\mathcal{H}_0]$ grows logarithmically with respect to the size of the data sample $\mathcal{D}$.

Consequently, the result in Preposition 1 gives rise to the following special properties:

1) **Score Consistency:** The more data we have - the more likely we converge to the set $\mathcal{I}_\ell(\mathcal{G}^*)$.
2) **Score Decomposability:** The score can be expressed (through algebraic manipulation) as a sum of family scores, which allows implementations to benefit from computational saving when performing a structure search.
3) **score-equivalence:** members from a set of I-equivalent structures will result in the same score as other members of the set.

The next section considers the prior, state space and search procedure to reconstruct a DIN.

D. Priors, State Space, and Search Procedure

In the previous sections we discussed a score and assemble that can be used to evaluate the quality of possible DINs between a collection of H-DBNs. We now consider the task of searching through different influence network structures and choosing a DIN which provides the highest score with respect to an assemble relation.

Many contributions in structure learning have been made over the last 60 years. These include attempts to reconstruct tree-like structures [9], Bayesian network structures [10], and general networks [11], [12].

a) Parameter Prior: Our implementation of discrete Dirichlet parameter priors follow those used by [13] and [14]. We make use of the BDe prior with score-equivalence and parameter independence (local and global) as essential properties [13].

b) Structure Prior: We use tree-based structure priors, where a score is calculated for every pair of H-DBNs in our collection of models. We then perform any polynomial time maximum weighted spanning tree (MWST-score) algorithm [15] to find the optimising structure.

For any restriction of the in-degree, learning a DIN structure between H-DBNs is NP-hard. Therefore, huristic search procedures are required. We are faced with a combinatorial optimisation problem to find direct influence between HBNs. We solve this problem by utilising a local search procedure.

We define a search space as a collection of candidate network structures; a scoring function, that we aim to maximize; the structure assemble, which associates our learned HDBN models; and finally, a search procedure which explores the search space. We have already discussed the structure score and assemble which leaves us to discuss the search space and elected procedure.

We explore a search space where each search state is a complete DIN. We connect our search space in terms of the following operations: *edge addition*, *edge removal*, and *edge reversal*. These computationally efficient operators provide a manageability small diameter of the search space (K^2) [5].

c) Search Procedure: [11] compared various search procedures including K2, local search, and simulated annealing. [11] show that local search offers the best time-accuracy tradeoff, unless a good ordering is known. In this study no such ordering is assumed. Therefore we employ a greedy hill-climbing local search procedure.

We pick an initial starting point DIN, $\mathcal{G}$, and calculate the score of $\mathcal{G}$ with respect to some structural assemble. We then consider all neighbours of $\mathcal{G}$ which are possible 1-step transformations given the predetermined operators, and compute their scores. Lastly, the change is applied which leads to the best improvement of the score.

The returned DIN, $\mathcal{G}$, from the iterative heuristic search procedure can either have reached a local optima or a plateau. In order to avoid these we use random restarts and a tabu list [16].

III. Results

Figure 2 shows the parameter and structure learning task performance for samples generated from a DIN with 4 HDBN models, 4 latent variables per HDBN network, 3 time-slices, 4 values per variable, 3 edges, a max in-degree of 2, and 2 observable variables.

Six learning tasks are compared using the KL-divergence of the learned to the true network. Each curve shows the averaged performance of 10 same sized data-sets, 3 time-slices (Ts), 2 observable variables (Obs), 4 latent variables (Var) learned with 10 EM iterations (EMit), 5 random restarts (RR), a tabu-list length of 5 (TL), and 20 structure search iterations (SSit). Each curve's description in Figure 2 is as follows:

1) **(Random)**, which learns parameters with a randomly generated direct influence graph structure, a maximum in-degree of 2 for each HDBN, and a Dirichlet parameter prior of 5;
2) **(Baseline)**, which learns both the parameters and structure representing each HDBN as a single variable (we plotted only the average of both *(Random)* and *(Baseline))*, Dirichlet parameter prior of 5, with a MWST-score tree structure prior, and combining each time-slice into one (cBS);
3) **(Low PPrior, Tree SPrior)**, a first setting of our method which learns both the parameters and structure with a Dirichlet parameter prior of 5, with a MWST-score tree structure prior;
4) **(High PPrior, Tree SPrior)**, a second setting of our method which learns both the parameters and structure with a Dirichlet parameter prior of 50, a MWST-score tree structure prior;
5) **(Low PPrior, No SPrior)**, a third setting of our method which learn both the parameters and structure with a Dirichlet parameter prior of 5, and no structure prior; and finally,
6) **(True Struc)**, which learns only the parameters given the correct network structure and a Dirichlet parameter prior of 5. The error bars shows +/- one standard deviation for the curves *(TrueStruc)* and *(High PPrior, Tree SPrior)*.

Of all methods provided, learning the direct influence structure using the proposed score-based approach with the HDBN BIC score and direct assemble, performs better than a randomly assigned structure. Figure 2 also suggests that using a MWST-score structure prior provides better performance towards learning the true network with a low prior; a higher Dirichlet parameter prior enables a more stable convergence to the true network; and treating H-DBNs as single variables may be better if there are fewer data instances available (in our analysis at about 85 samples our method does better). The complications of the search procedure could be derived from the latent components of each HDBN, which are the only means of transferring information between the HDBN model variables.

IV. Conclusion

This paper provided the first score-based structure learning algorithm to learn the network structure and parameters (distribution) of a *DIN structure* between a collection of processes. Figure 2 suggests that it is not significantly harder to recover both the structure and parameters than just the parameters, which is reason for optimism.

Future work one can investigate the use of operators able to traverse the search space in larger steps [17]. An advantage of doing this can avoid local optima, however care must be taken to avoid cyclic iterations due to disregarding steep gradients. Finally, A more detailed analysis on parameter tuning is necessary to optimise the performance of our approach.

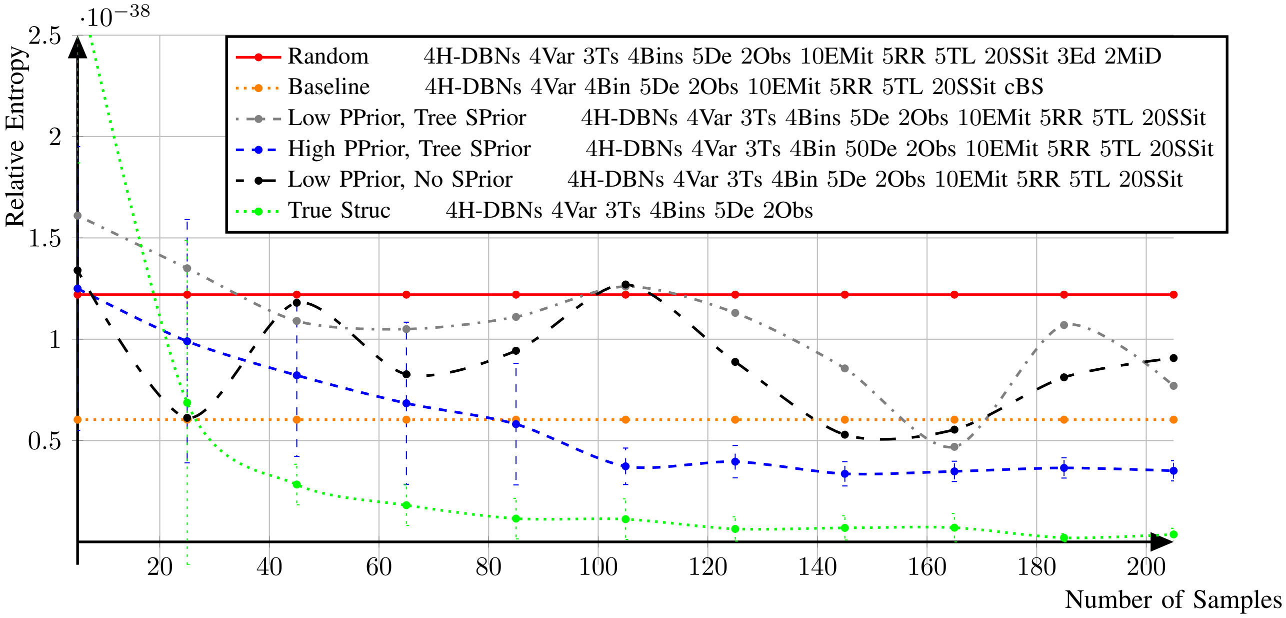

Figure 2: The performance of parameter and structure learning tasks for instances generated from a DIN with 4 HDBN models, 4 latent variables per HDBN network (Var), 3 time-slices (Ts), 4 values per variable (Bins), 3 edges (Ed), a max in-degree of 2 (MiD), and 2 observable variables (Obs).

Acknowledgements: This work is based on the research supported in part by the National Research Foundation of South Africa (Grant number: 121835).

References

[1] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[2] D. J. Miller and G. Judge, “Information recovery in a dynamic statistical markov model,” *Econometrics*, vol. 3, no. 2, pp. 187–198, 2015.

[3] P. A. Valdes-Sosa, A. Roebroeck, J. Daunizeau, and K. Friston, “Effective connectivity: influence, causality and biophysical modeling,” *Neuroimage*, vol. 58, no. 2, pp. 339–361, 2011.

[4] D. Commenges and A. Gégout-Petit, “A general dynamical statistical model with causal interpretation,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 719–736, 2009.

[5] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[6] F. Bacchus and A. Grove, “Graphical models for preference and utility,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 3–10.

[7] R. Ajoodha and B. Rosman, “Tracking influence between naïve bayes models using score-based structure learning,” in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*. IEEE, 2017, pp. 122–127.

[8] G. Schwarz *et al.*, “Estimating the dimension of a model,” *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[9] C. Chow and C. Liu, “Approximating discrete probability distributions with dependence trees,” *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[10] G. F. Cooper and E. Herskovits, “A bayesian method for the induction of probabilistic networks from data,” *Machine learning*, vol. 9, no. 4, pp. 309–347, 1992.

[11] D. Chickering, D. Geiger, and D. Heckerman, “Learning bayesian networks: Search methods and experimental results,” in *proceedings of fifth conference on artificial intelligence and statistics*, 1995, pp. 112–128.

[12] W. Buntine, “Theory refinement on bayesian networks,” in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1991, pp. 52–60.

[13] D. Heckerman and D. Geiger, “Learning bayesian networks: a unification for discrete and gaussian domains,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 274–284.

[14] D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, and R. G. Cowell, “Bayesian analysis in expert systems,” *Statistical science*, pp. 219–247, 1993.

[15] B. Moret and H. Shapiro, “An empirical analysis of algorithms for constructing a minimum spanning tree,” *Algorithms and Data Structures*, pp. 400–411, 1991.

[16] F. Glover and M. Laguna, *Tabu Search*. Springer, 2013.

[17] A. Moore and W.-K. Wong, “Optimal reinsertion: A new search operator for accelerated and more accurate bayesian network structure learning,” in *ICML*, vol. 3, 2003, pp. 552–559.