Tracking Influence between Naïve Bayes Models using Score-Based Structure Learning

Ritesh Ajoodha and Benjamin Rosman

***Abstract*—Current structure learning practices in Bayesian networks have been developed to learn the structure between observable variables and learning latent parameters independently. One exception establishes a variant of EM for learning the structure of Bayesian networks in the presence of incomplete data [1]. However, no method has demonstrated learning the influence structure between latent variables that describe (or are learned from) a number of observations. We present a method that learns a set of naïve Bayes models (NBMs) independently given a partitioned set of observations, and then attempts to track the high-level influence structure between every NBM. The latent parameters of each model are then relearned to fine-tune the influence distribution between models for density estimation of new observations. Experimental results are provided which demonstrate the effectiveness of our non-parametric method. Applications of this method include knowledge discovery and density estimation in situations where we do not fully observe characteristics of the environment.**

***Index Terms*—Score-based structure learning, naïve Bayes models, Bayesian networks, structure scores, Bayesian information criterion, heuristic search, greedy hill-climbing, expectation maximisation.**

I. Introduction

Learning probability distributions using graphical models has been a major accomplishment. In these graphical models the joint probability distribution was described as influence between random variables encoded as a directed acyclic graph (DAG) which embed independence assertions [2]. These developments gave rise to the notion of a Bayesian network which translate these independent assertions into a DAG that encodes a joint probability distribution [3]. Bayesian networks span a range of applications including general diagnostic systems [4]; event forecasting [5]; machine vision [6]; and even music classification [7].

Perhaps the most common method of parameter estimation in Bayesian networks is maximum likelihood estimation (MLE), which optimises the likelihood function for complete data. Unfortunately, we are not always given complete data and often instead given a set of observations that describe a latent phenomenon (not explicitly observed). Recovering the influence structure from incomplete data, that is induced by a set of observations, appears in many real-world applications where we wish to perform density estimation, without overfitting, or learn structure between latent characteristics of an environment for knowledge discovery (e.g. learning the influence of traffic on roads in an area, or learning how proteins interact in a cell).

In this paper we attempt to recover the structure of influence between naïve Bayes models (NBMs). That is, can we recover the ground truth influence structure or distribution which gave rise to the data. Figure 1 shows an example of the influence structure between a set of NBMs (section II-A). Each latent variable may describe class values that represent the set of observations.

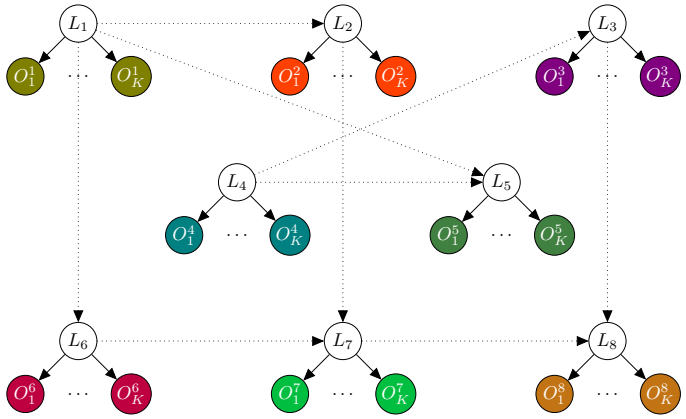

Fig. 1: A graphical depiction of the influence structure between several NBMs. Each set of observations for latent variable L_i is denoted as $O_1^i, \ldots, O_K^i$. The solid lines indicate the conditional independence assumptions of each NBM, and the dotted lines indicate the high-level structure of influence between each NBM.

A popular method to structure discovery, in observable data, is score-based structure learning, where we use a scoring metric to search for the most suitable structure relative to the data. Most popular structure scores are variations on the likelihood score which calculates the probability of the data given a potential structure. In observable data the decomposability of the likelihood score, which is the ability to represent the score as a sum of family scores, allows for efficient learning procedures and significant computational saving.

However, in incomplete data, the likelihood score is not decomposable and we have to perform inference to evaluate it. This forces us to use non-linear optimisation techniques such as EM or gradient ascent [8]. Furthermore, local changes to the network can affect other parts of the network, which makes learning with incomplete data all the more difficult.

The novelty and intuition of our approach is to learn the optimal influence structure in layers. We firstly learn a set of independent models, and thereafter, optimise a structure

R. Ajoodha is with the Department of Computer Science and Applied Mathematics, The University of the Witwatersrand, Johannesburg, South Africa. e-mail: ritesh.ajoodha@wits.ac.za.

B. Rosman is with the Council for Scientific and Industrial Research as well as the Department of Computer Science and Applied Mathematics, The University of the Witwatersrand, South Africa. e-mail: brosman@csir.co.za.

score over possible structural configurations. Since the search for the optimal structure is done using complete data we can take advantage of efficient learning procedures and significant computational saving from the structure learning literature. We provide the following contributions. (a) The notion of influence between the NBM representation; (b) an extension of the traditional BIC score for NBMs rather than random variables; (c) introduce a complete algorithm to recover the influence structure between NBMs; (d) provide empirical evidence for the effectiveness of our method.

This paper is structured as follows. We begin by reviewing current literature on parameter estimation for complete and incomplete data in section II; section III reviews current work in structure learning; section IV provides the specification of the structure score for this problem; section V presents the empirical results which show the effectiveness of our method; and finally, section VI discusses various applications of influence networks and future work.

II. Background

A Bayesian network is a directed acyclic graph (DAG) whose nodes represent random variables and whose edges represent the influence of one variable over another. A Bayesian network structure is often established as a set of independence assertions between these random variables that encode a joint distribution in a compact way [9].

In this section we explore preliminary work in parameter estimation. More specifically, we will explore the definition and representation of a NBM (section II-A); learning parameters for observable data (section II-B); and finally, learning parameters for missing data (section II-C).

A. The Naïve Bayes Model

Perhaps the simplest example of a Bayesian model is the naïve Bayes model (NBM) which has been traditionally and successfully used by many expert systems [10].

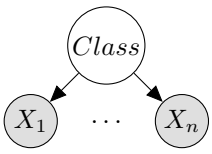

Fig. 2: An illustration of the NBM.

The NBM predefines a finite set of mutually exclusive classes. Each instance can fall into one of these classes, this is represented as a latent class variable. The model also poses some observed set of features $X_1, \ldots, X_n$. The assumption is that all of the features are conditionally independent given the class label of each instance. That is $\forall \mathbf{i}(\mathbf{X_i} \perp\!\!\!\perp \mathbf{X_{i'}} \mid \mathbf{C})$, where $X_{i'} = \{X_1, \ldots, X_n\} - \{X_i\}$

Figure 2 presents the Bayesian network representation of the NBM. The joint distribution of the NBM factorises compactly as a prior probability of an instance belonging to a class, $P(C)$, and a set of conditional probability distributions (CPDs) which indicate the probability of a feature given the class. We can state this distribution more formally as

$$P(C, X_1, \ldots, X_n) = P(C) \prod_{i=1}^{n} P(X_i|C).$$

The NBM remains a simple, yet highly effective, compact, and high-dimensional probability distribution that is often used for classification problems.

B. Learning with Observable Data

Perhaps the simplest parameter learning tool is *maximum likelihood estimation* (MLE) from a set of observations. MLE is foundational to many parameter learning problems and much work has been dedicated to its development [11]. An alternative to MLE for learning the parameters of variables in a Bayesian network is Bayesian estimation (BE). BE follows the Bayesian paradigm which views any event that has uncertainty as a random variable with a distribution over it.

Suppose we have a data-set, $\mathcal{D} = \{\xi_1, \ldots, \xi_M\}$, where each instance, ξ_m, contains only one observation x, then we can express the joint distribution of each observation (as well as the parameter θ which generates the data) by using the chain rule for Bayesian networks as

$$P(x[1], \ldots, x[M], \theta) = P(x[1], \ldots, x[M]|\theta)P(\theta),$$

which gives us the prior over θ and the probability of each instance given the parameter θ,

$$P(x[1], \ldots, x[M], \theta) = P(\theta) \prod_{i=1}^{M} P(x[i]|\theta).$$

We note the similarities to MLE in the above expression with the additional prior probability over θ. We can express the posterior of this prior, given the data, using Bayes rule [12]:

$$P(\theta|x[1], \ldots, x[M]) = \frac{P(x[1], \ldots, x[M]|\theta)P(\theta)}{P(x[1], \ldots, x[M])}.$$

There are many choices for a prior distribution, but a common choice is the Dirichlet prior [13] which is characterised by a set of hyper-parameters $(\alpha_1, \ldots, \alpha_k)$.

C. Learning with Missing Data

We now consider a much more difficult problem of learning parameters from missing data. Latent variables provide a sparse parameterisation of a distribution and can be used to aggregate observable variables.

In observable data we can optimise the likelihood of the parameters, given the data, using MLE. Unfortunately, the likelihood function for missing data could have multiple optima which we cannot easily search for. A common strategy to optimise the likelihood function is Expectation Maximisation (EM) which attempts to learn both the missing data and the parameters iteratively [8]. The EM algorithm generalises several algorithms including the Baum-Welch algorithm used for learning HMMs [14]. The general skeleton of the EM algorithm is outlined in Algorithm 1.

Line 4 of Algorithm 1 is called the E-step. In this step we perform Bayesian inference to infer the data given the parameters. That is, for each instance, $\xi[\mathbf{m}]$, and each family,

variable X and parent-set $\mathbf{U}$, we compute $P(X, \mathbf{U}|\xi[m], \theta^t)$, where θ^t is the current setting of the parameters at iteration t. We now can compute the expected sufficient statistics, $\hat{M}$, for each combination of values $(x, \mathbf{u})$ per family. We can express the expected sufficient statistics for $(x, \mathbf{u})$ at time-slice t as

$$\hat{M}_{\theta^t}[x, \mathbf{u}] = \sum_{m=1}^{M} P(x, \mathbf{u}|\xi[m], \theta^t).$$

Algorithm 1 Expectation Maximisation

1: **procedure** EXPECTATION-MAXIMISATION($\langle \mathcal{B}_0, \mathcal{B}_{\rightarrow} \rangle, \mathcal{D}$)
2: Pick a starting point for the parameters.
3: **for** until convergence **do**
4: Complete the data using the current parameters
5: Estimate the parameters relative to data completion
6: **end for**
7: **return** Data and parameters
8: **end procedure**

Line 5 of Algorithm 1 is called the M-step. In the M-step we treat the expected sufficient statistics, $\hat{M}$, as real sufficient statistics and then use MLE or BE. The EM algorithm is guaranteed to monotonically improve the log-likelihood of the parameters relative to the data at each iteration.

III. Related Work

Score-based structure learning requires the definition of a hypothesis space of potential network structures; defines a structure score which gauges how well the model fits the observed training data; and finally, attempts to find the highest scoring structure as an optimisation problem. However, given that the search space is super-exponential in size, we resort to heuristic techniques.

Score-based structure learning is our first choice approach since it (a) considers the complete influence structure between models as a state in the search space; (b) preserves basic score properties to allow for feasible computations; and (c) it provides a clear indication of the independence assertions between the concerned structures relative to the data. In this section we review related traditional score-based structure learning practices. We begin by discussing structure scores in section III-A, and then move to the much more difficult problem of learning graph structures in section III-B.

A. Structure score

There are several choices of structure scores geared at evaluating the likelihood of a particular structure given the fit to data. A well-known choice is that of the likelihood score which maximises the likelihood (or log-likelihood in practice) of the structure relative to the data. We can express this as the MLE, $\hat{\theta}$, given a particular graph structure, $\mathcal{G}$, relative to the data, $\mathcal{D}$. This gives us the likelihood score denoted as $score_L = \ell((\hat{\theta}, \mathcal{G}) : \mathcal{D})$. In other words, if we are presented with a particular graph structure we will find the maximum likelihood estimate for the parameters of that graph with respect to the data. This can be expressed more generally as a relative cost of adding an edge between variables in a graph structure.

The likelihood score decomposes as the number of instances multiplied by the mutual information, ($\mathbf{I}_{\hat{P}}$), between each family of variables, minus the entropy of each variable that is independent of the structure. More formally,

$$score_L(\mathcal{G}, \mathcal{D}) = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(X_i; \mathbf{Pa}_{X_i}^{\mathcal{G}}) - M \sum_{i=1}^{n} \mathbf{H}_{\hat{P}}(X_i)$$

where $\mathbf{I}_{\hat{P}}(X; Y) = \sum_{x,y} P(x, y) \log \frac{P(x,y)}{P(x)P(y)}$; $\mathbf{H}_{\hat{P}}(X) = -\sum_x P(x) \log P(x)$; and $\mathbf{Pa}_{X_i}^{\mathcal{G}}$ is the parents of X_i relative the graph structure $\mathcal{G}$.

The fact that the most complicated network is always preferred by the likelihood score, poses a significant over-fitting problem. This is usually overcome by regulating the hypothesis space or penalising structural complexity.

The Bayesian information criterion (BIC), is a popular choice for trading-off model complexity and fit to data. The BIC score consists of two terms: (i) the first describes the fit of the hypothesised structure to the data, usually the likelihood function $score_L = \ell((\hat{\theta}, \mathcal{G}) : \mathcal{D})$; and (ii) the second is a penalty term for complex networks. More formally the BIC score is given as

$$score_{BIC} = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} DIM[\mathcal{G}],$$

where M is the number of training instances and $DIM[\mathcal{G}]$ is the number of independent parameters in the network. We note that the entropy component of the likelihood term is negligible since it does not depend on the selected structure. This observation allows us to rewrite the BIC score as

$$score_{BIC} = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(X_i; \mathbf{Pa}_{X_i}^{\mathcal{G}}) - \frac{\log M}{2} DIM[\mathcal{G}].$$

The BIC score has the following properties. (a) As we increase the number of samples the emphasis moves from model complexity to the fit to data. In other words, as we obtain more data we are more likely to consider more complicated structures. (b) As the BIC score acquires more data it approaches the true structure (or one which is i-equivalent, that is a structure which makes the same independence assumptions). (c) The BIC score gives the same score for members of the same i-equivalence class, that is, different structures which encode the same independent assumptions.

B. Learning Graph Structures

In the case of learning general graph-structures the structure learning problem's complexity increases significantly [12]. More formally [12],

Theorem 1. *For any dataset, $\mathcal{D}$, and decomposable structure score, score, the problem of finding the maximum scoring network, that is,*

$$\mathcal{G}^* = \arg \max_{\mathcal{G} \in \mathcal{G}_{\mathbf{d}}} score(\mathcal{G} : \mathcal{D}),$$

is NP-Hard for any $d \geq 2$, where $\mathcal{G}_{\mathbf{d}} = \{\mathcal{G} : \forall i, |Pa_{X_i}^{\mathbf{G}}| \leq d\}$.

In other words, finding the maximal scoring network structure with at most d parents for each variable is NP-hard for any d greater than 2. This is because of the super-exponential search space that one has to traverse to obtain the maximal network. The above result might be discouraging. However, using local search procedures we are able to provide a solution using heuristic hill-climbing [15].

There are two main design choices that one needs to make when using a local structure search procedure: (i) the choice of search operators and (ii) search procedure.

Search operators are local steps to traverse the search space. Common choices for local search operators are edge addition, reversal, and deletion.

A common choice for a search procedure is greedy hill-climbing. The greedy hill-climbing approach starts by selecting a prior network. The prior network could be an empty structure; a best tree structure; a random structure; or a structure elicited by an expert. From this prior network we iteratively try to improve the structure's score by utilising search operators. In greedy hill-climbing we always apply the change that improves the score until no improvement can be made.

IV. Learning the Influence Between NBMs

In this paper we provide the first score-based structure learning algorithm and analysis of learning the influence structure between a set of NBMs (with incomplete data). The influence structure encodes independence assumptions and factorises a joint distribution between a finite set of NBMs.

We assume that the data generated from the ground truth influence structure has the following form: $\mathcal{D}_{\mathbb{I}} = \{\mathcal{D}_{\mathcal{B}^1}, \ldots, \mathcal{D}_{\mathcal{B}^k}\}$, where $\mathcal{D}_{\mathcal{B}^i} = \{\xi_1, \ldots, \xi_{\mathcal{M}}\}$ is a set of M instances for NBM $\mathcal{B}^i$. Each ξ_j is generated independently and identically distributed from an underlying distribution, $P^*(\mathbb{I}^{\mathcal{G}})$, where $\mathbb{I}$ is a Bayesian network which represents the influence structure $\mathcal{G}$.

$\mathbb{I}^{\mathcal{G}}$ contains a distribution between a set of NBMs, $\mathcal{B}^1, \ldots, \mathcal{B}^k$, with the independence assumptions specified by $\mathcal{I}_\ell(\mathbb{I}^{\mathcal{G}})$. We further assume that $P^*(\mathbb{I}^{\mathcal{G}})$ is induced by another model, $\mathcal{G}^*(\mathbb{I}^{\mathcal{G}})$, which we refer to as the *ground truth* structure. We will evaluate our model by recovering the local independence assertions and joint distribution in $\mathcal{G}^*(\mathbb{I}^{\mathcal{G}})$, denoted $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^{\mathcal{G}}))$, by only observing $\mathcal{D}_{\mathbb{I}^{\mathcal{G}}}$. The architecture of the proposed algorithm is given by Figure 3. We (i) learn each NBM independently (using EM); (ii) compute the structure score of the complete influence network (using a scoring function); (iii) relearn the parameters for the model (with the new independence assertions between NBMs) which gives us the candidate network (iv); (v) perform an operation (edge addition, reversal, or removal) to try to improve the network fit to data; Steps (ii), (iii), (v) are repeated until we can not improve the score for the structure. We then select the best network (iv). In the next section we develop the structure score for influence network.

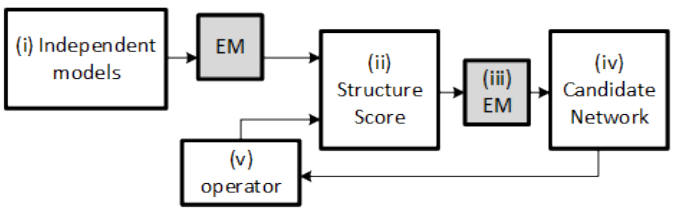

Fig. 3: The architecture of the proposed algorithm.

A. Structure Score

Intuitively we would like to measure that if a relation that describes influence between two NBMs is preferred by the data, then we would get more information from having this relation than without having it [12].

Consider Figure 4 which illustrates two scenarios between two NBMs, $\mathcal{B}_0 = (\mathcal{G}_{\mathcal{B}_0}, P_{\mathcal{B}_0})$ and $\mathcal{B}_1 = (\mathcal{G}_{\mathcal{B}_1}, P_{\mathcal{B}_1})$, where $\mathcal{G}_{\mathcal{B}_i}$ and $P_{\mathcal{B}_1}$ is the structure and probability distribution for $\mathcal{B}_i$ respectively.

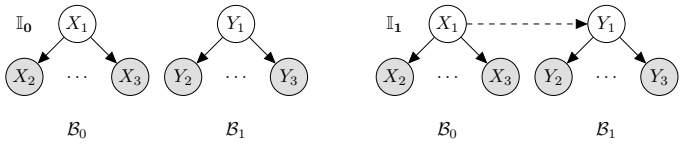

Fig. 4: Two influence structures between two NBMs.

$\mathcal{B}_0$ and $\mathcal{B}_1$ encode the local independence assumption $\mathcal{I}_\ell(\mathcal{G}_{\mathcal{B}_0}) = \{X_2 \perp\!\!\!\perp X_3 |\ X_1\}$ and $\mathcal{I}_\ell(\mathcal{G}_{\mathcal{B}_1}) = \{Y_2 \perp\!\!\!\perp Y_3 |\ Y_1\}$ respectively. Each of these scenarios in Figure 4 can be described as an influence network, denoted $\mathbb{I}_0 = (\mathcal{G}_0, P_{\mathbb{I}_0})$ and $\mathbb{I}_1 = (\mathcal{G}_1, P_{\mathbb{I}_1})$ respectively, where $\mathcal{G}_i$ denotes the structure of the influence network $\mathbb{I}_i$ with probability distribution $P_{\mathbb{I}_i}$. We assume that influence between networks, which may describe events, flows at a level of abstraction constructed by observable variables and not directly from the observations themselves (since the observations are dependent on the time granularity).

Selecting an influence network, either $\mathbb{I}_0$ or $\mathbb{I}_1$, requires us to establish which structure, either $\mathcal{G}_0$ or $\mathcal{G}_1$, gives us a stronger likelihood to the data. Let us express the preferability of a particular structure more formally. The log-likelihood of $\mathcal{G}_0$ relative to the data, denoted $\text{score}_L(\mathcal{G}_0 : \mathcal{D})$, can be expressed as:

$$\sum_{m=1}^{M} (\log \hat{\theta}_{x_1[m]} + \log \hat{\theta}_{x_2[m]|x_1[m]} + \log \hat{\theta}_{x_3[m]|x_1[m]} + \underline{\log \hat{\theta}_{y_1[m]}} + \log \hat{\theta}_{y_2[m]|y_1[m]} + \log \hat{\theta}_{y_3[m]|y_1[m]}), \tag{1}$$

and the log-likelihood score of $\mathcal{G}_1$ relative to the data, denoted $\text{score}_L(\mathcal{G}_1 : \mathcal{D})$, can be expressed as:

$$\sum_{m=1}^{M} (\log \hat{\theta}_{x_1[m]} + \log \hat{\theta}_{x_2[m]|x_1[m]} + \log \hat{\theta}_{x_3[m]|x_1[m]} + \underline{\log \hat{\theta}_{y_1[m]|x_1[m]}} + \log \hat{\theta}_{y_2[m]|y_1[m]} + \log \hat{\theta}_{y_3[m]|y_1[m]}), \tag{2}$$

where $\hat{\theta}_{x_i}$ is the MLE for $P(x_i)$ and $\hat{\theta}_{y_j|x_i}$ is the MLE for $P(y_j|x_i)$.

To intuitively express the trade-off of using one influence structure, between these NBMs, over the other, we would like

to find which influence structure maximises the likelihood to the data. We can express this as the difference between the log-likelihood score of each model relative to the data as follows:

- if we have $\text{score}_L(\mathbb{I}_1^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) - \text{score}_L(\mathbb{I}_0^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) > 0$, then we would prefer the structure $\mathbb{I}_1^{\mathcal{G}}$;
- if $\text{score}_L(\mathbb{I}_1^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) - \text{score}_L(\mathbb{I}_0^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) < 0$, then we would prefer the structure $\mathbb{I}_0^{\mathcal{G}}$;
- finally, if $\text{score}_L(\mathbb{I}_1^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) - \text{score}_L(\mathbb{I}_0^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) = 0$, then either structure will do since both give us the same likelihood relative to the data.

By subtracting Equation 2 from Equation 1 we can express the difference of computing the log-likelihood scores for either influence structure over the two NBMs, denoted, $\text{score}_L(\mathbb{I}_1^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) - \text{score}_L(\mathbb{I}_0^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}})$, as

$$\sum_{m=1}^{M} (\log \hat{\theta}_{y_1[m]|x_1[m]} - \log \hat{\theta}_{y_1[m]}). \tag{3}$$

We can convert the summation, in Equation 3, to summing over values rather than over data instances. Thus we can representing each term by its respective sufficient statistics to obtain

$$\sum_{x_1,y_1} M[x_1, y_1] \log \hat{\theta}_{y_1|x_1} - \sum_{y_1} M[y_1] \log \hat{\theta}_{y_1}. \tag{4}$$

The first summation in Equation 4 expresses the summation over all parameters of values, denoted $Val(Y_1)$ given $Val(X_1)$, multiplied by the number of times these values occur in the data. We can more clearly express this as an empirical distribution $\hat{P}(x_1, y_1)$ which is expressed in the training data $\mathcal{D}_{\mathbb{I}}$. The sufficient statistic $M[x_1, y_1]$ is equal to the number of data instances multiplied by the empirical joint distribution, $M\hat{P}(x_1, y_1)$. Similarly we can state that $M[y_1] = M\hat{P}(y_1)$; $\hat{\theta}_{y_1|x_1} = \hat{P}(y_1|x_1)$, and $\hat{\theta}_{y_1} = \hat{P}(y_1)$.

If we express Equation 4 in terms of the empirical distribution, the difference in the score becomes

$$\sum_{x_1,y_1} M\hat{P}(x_1, y_1) \log \hat{P}(y_1|x_1) - \sum_{y_1} M\hat{P}(y_1) \log \hat{P}(y_1). \tag{5}$$

Both summations in Equation 5 contain the number of samples M which is independent of type of values found in the data and thus M can be extracted from the summation.

Both summations could have been condensed into one if they were summed over the same values. We can artificially insert the sum over x_1 in the second summation of Equation 5 since $\sum_{x_1} \hat{P}(x_1, y_1) = \hat{P}(y_1)$. Thus we get

$$M(\sum_{x_1,y_1} \hat{P}(x_1, y_1) \log \hat{P}(y_1, x_1) - \sum_{x_1,y_1} \hat{P}(x_1, y_1) \log \hat{P}(y)). \tag{6}$$

There are two more manipulations that we can exploit in Equation 6 to condense the difference of the scores further. Firstly, the term $\hat{P}(y_1|x_1)$ can be rewritten as $\frac{\hat{P}(x_1,y_1)}{\hat{P}(x_1)}$ using Bayes rule [12]; and secondly, both summations in Equation 6 are of the same form and the term $\hat{P}(x_1, y_1)$ is a common term in each summation. Therefore, using the subtraction rule for logarithms we can condense the difference of the two scores as

$$M \sum_{x_1,y_1} \hat{P}(x_1, y_1) \log \frac{\hat{P}(y_1, x_1)}{\hat{P}(y_1)\hat{P}(x_1)}. \tag{7}$$

The summation in Equation 7 is called the *mutual information* of $\mathcal{B}_0$ and $\mathcal{B}_1$ since it measures the average distance between the joint distribution, of $\mathcal{B}_0$ and $\mathcal{B}_1$, relative to if their distribution was a product of marginally independent models. We denote the mutual information of the two Bayesian networks as

$$\text{score}_L(\mathbb{I}_1^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) - \text{score}_L(\mathbb{I}_0^{\mathcal{G}} : \mathcal{D}_{\mathbb{I}}) = M.\mathbf{I}_{\hat{P}}(\mathcal{B}_0; \mathcal{B}_1). \tag{8}$$

Thus the BIC score for NBMs decomposes as

$$score_{BIC} = M \sum_{i=1}^{n} \mathbf{I}_{\hat{P}}(\mathcal{B}_i; \mathbf{Pa}_{\mathcal{B}_i}^{\mathcal{G}}) - \frac{\log M}{2} DIM[\mathcal{G}].$$

The BIC score between Bayesian models has the following properties. (a) As we increase the number of samples the emphasis moves from model complexity to the fit to data. In other words, as we obtain more data we are more likely to consider more complicated structures. (b) As the BIC score acquires more data it approaches the true structure (or one which is i-equivalent). (c) The BIC score gives the same score for members of the same i-equivalence class.

Note the difference between the traditional mutual information for variables in section II and the mutual information between Bayesian models in Equation 7. In this example it may seem fairly similar but depending on the number of latent variables in the model, and the structural properties which are expressed in the model, the mutual information over models generally aggregates structural correlations and similarities over all of the variables in each Bayesian model.

V. Empirical Results

Figure 5 shows the performance of four parameter or structure learning tasks to recover the ground truth's distribution. The y-axis is the relative entropy to the true distribution, $P^*(\mathbb{I}^{\mathcal{G}})$, and the x-axis represents the number of training samples. The ground truth structure, $\mathcal{G}^*(\mathbb{I}^{\mathcal{G}})$, had 24 edges, 3 bin values per variable, a Dirichlet prior of 5, 5 observations per NBM, 10 NBMs, and a max in-degree of 3. The four learning tasks are: *random structure*, where a 'random' structure is generated with knowledge of the ground truth's parameters; *No structure*, where no conditional independence assertions are present between models; *Learned structure*, where we simultaneously estimate the parameters and structure between models using a tabu list [16] of length 5 and 5 random restarts; and finally, *True structure*, where we are given the true structure between models and attempt to learn the parameters. All latent variables were learned using 10 EM iterations and achieved a 60-82% accuracy validated against the ground truth.

The relative entropy of an i-equivalent structure to $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^{\mathcal{G}}))$ will still get close to recovering the true distribution, $P^*(\mathbb{I}^{\mathcal{G}})$. As we see our method tends to correctly recover the distribution between each NBM compared to random guesses and over mutually independent models, except for when we have very little training instances.

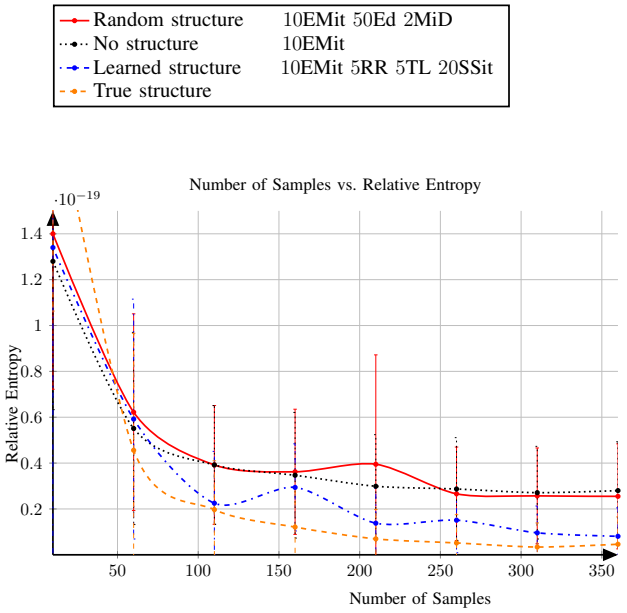

Fig. 5: The performance of parameter and structure learning tasks for instances generated from an influence network between 10 NBMs.

VI. Conclusion and Future Work

In this paper, we proposed the first method to learn the influence structure between a set of NBMs. The main idea of our approach is (a) learning each NBM independently (using EM); (b) expressing the problem as structure learning in the case of complete data (which can be solved efficiently using current literature); and finally, (c) using EM to fine-tune the latent parameter estimates for each independence assertion introduced. Steps (b) and (c) are used together to select the optimal structure.

The significance of learning an influence network depends on our objective. If one is attempting to discover exactly the ground truth network structure which involves stating precisely $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^\mathcal{G}))$, then we should concede that many *perfect maps* [12] for $P^*(\mathbb{I}^\mathcal{G})$ can be achieved from $\mathcal{D}_\mathbb{I}$. It is understood that recognising $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^\mathcal{G}))$ from $\mathcal{G}^*(\mathbb{I}^\mathcal{G})$'s set of structures, which give the same fit to the data (I-equivalent structures), is *not identifiable* from $\mathcal{D}_\mathbb{I}$ since each I-equivalent structure produces the same likelihood for $\mathcal{D}_\mathbb{I}$. Therefore, if our goal is knowledge discovery, we should instead try to recover $\mathcal{G}^*(\mathbb{I}^\mathcal{G})$'s i-equivalence class. This is difficult as data sampled from $P^*(\mathbb{I}^\mathcal{G})$ does not perfectly and uniquely reconstruct the independence assumptions of $\mathcal{G}^*(\mathbb{I}^\mathcal{G})$.

Alternatively, one could also attempt to learn an influence network for *density estimation*, i.e. to estimate a statistical model of the underlying distribution $P^*(\mathbb{I}^\mathcal{G})$. Such a model can be used to reason about new data instances. On the one hand, if we capture more independence assertions than those specified in $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^\mathcal{G}))$, we could still capture $P^*(\mathbb{I}^\mathcal{G})$ using some setting of our recovered networks parameters. However, our selection of more independence assumptions, rather than fewer in $\mathcal{I}_\ell(\mathcal{G}^*(\mathbb{I}^\mathcal{G}))$, could result in *data fragmentation*. On the other hand, selecting too few edges will result in not capturing the true distribution $P^*(\mathbb{I}^\mathcal{G})$, but will however provide a sparse structure that avoids data fragmentation. Generally, the latter case is preferred in density estimation since it provides better generalisation to new instances through a sparser representation [12].

The following future work can be explored. (a) We can consider using operators which take much larger steps in the search space making the search procedure less susceptible to local optima [12]. (b) Other aspects of influence can be explored by extending the extent that models can influence each other. (c) More sophisticated techniques can be employed to explore the scope of influence in the temporal setting.

Acknowledgment

The first author gratefully acknowledges the support of the *NRF Scarce Skills Doctoral Scholarship*, and *the Post-Graduate Merit Award Scholarship* for Doctoral Studies.

References

[1] N. Friedman *et al.*, "Learning belief networks in the presence of missing values and hidden variables," in *ICML*, vol. 97, 1997, pp. 125–133.
[2] M. Kevin, "Machine learning: a probabilistic perspective," 2012.
[3] T. S. Verma and J. Pearl, "Causal networks: Semantics and expressiveness," *arXiv preprint arXiv:1304.2379*, 2013.
[4] S. Andreassen, M. Woldbye, B. Falck, and S. K. Andersen, "Munin: A causal probabilistic network for interpretation of electromyographic findings," in *Proceedings of the 10th international joint conference on Artificial intelligence-Volume 1*. Morgan Kaufmann Publishers Inc., 1987, pp. 366–372.
[5] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.
[6] T. O. Binford, T. S. Levitt, and W. B. Mann, "Bayesian inference in model-based machine vision," *arXiv preprint arXiv:1304.2720*, 2013.
[7] R. Ajoodha, R. Klein, and B. Rosman, "Single-labelled music genre classification using content-based features," in *IEEE proceedings, Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2015*, Nov 2015, pp. 66–71.
[8] S. R. Tembo, S. Vaton, J.-L. Courant, and S. Gosselin, "A tutorial on the em algorithm for bayesian networks: application to self-diagnosis of gpon-ftth networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*. IEEE, 2016, pp. 369–376.
[9] J. Pearl, "Probabilistic reasoning in intelligent systems. palo alto," *Morgan Kaufmann. PEAT, J., VAN DEN BERG, R., & GREEN, W.(1994). Changing prevalence of asthma in australian children. British Medical Journal*, vol. 308, pp. 1591–1596, 1988.
[10] D. M. Khairina, S. Maharani, H. R. Hatta *et al.*, "Decision support system for admission selection and positioning human resources by using naive bayes method," *Advanced Science Letters*, vol. 23, no. 3, pp. 2495–2497, 2017.
[11] M. H. DeGroot and M. J. Schervish, *Probability and statistics*. Addison-Wesley,, 2012.
[12] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
[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] L. R. Welch, "Hidden markov models and the baum-welch algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, pp. 10–13, 2003.
[15] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine learning*, vol. 65, no. 1, pp. 31–78, 2006.
[16] F. Glover and M. Laguna, *Tabu Search**. Springer, 2013.