Modelling regimes with Bayesian network mixtures

Marcus Bendtsen and Jose M. Peña

Modelling regimes with Bayesian network mixtures

Marcus Bendtsen and Jose M. Peña
marcus.bendtsen@liu.se | jose.m.pena@liu.se
Linköping University, Department of Computer and Information Science, Sweden

Abstract

Bayesian networks (BNs) are advantageous when representing single independence models, however they do not allow us to model changes among the relationships of the random variables over time. Due to such *regime changes*, it may be necessary to use different BNs at different times in order to have an appropriate model over the random variables. In this paper we propose two extensions to the traditional hidden Markov model, allowing us to represent both the different regimes using different BNs, and potential driving forces behind the regime changes, by modelling potential dependence between state transitions and some observable variables. We show how expectation maximisation can be used to learn the parameters of the proposed model, and run both synthetic and real-world experiments to show the model's potential.

Keywords

Bayesian networks, hidden Markov models, regimes, algorithmic trading.

1 INTRODUCTION

Introduced by Judea Pearl [1], Bayesian networks (BNs) consist of two components: a qualitative representation of independencies amongst random variables through a directed acyclic graph (DAG), and a quantification of certain marginal and conditional probability distributions, so as to define a full joint probability distribution over the random variables. A feature of BNs, known as the local Markov property, implies that a variable is independent of all other non-descendant variables given its parent variables, where the relationships parent and descendant are defined with respect to the DAG of the BN. Let $\mathbf{X}$ be a set of random variables in a BN, and let $pa(X_i)$ represent the set of variables that consists of the parents of variable $X_i \in \mathbf{X}$, then the local Markov property allows us to factorise the joint probability distribution according to Equation 1.

$$p(\mathbf{X}) = \prod_{X_i \in \mathbf{X}} p(X_i | pa(X_i)) \quad (1)$$

From Equation 1 it is evident that the independencies represented by the DAG allow for a representation of the full joint distribution via smaller marginal and conditional probability distributions, thus making it easier to elicit the necessary parameters, and allowing for efficient computation of posterior probabilities. For a full treatment of BNs, please see [2, 3, 1].

While a BN has advantages when representing a single independence model, it does not allow us to model changes of the independencies amongst the modelled variables over time. One reason why we would take such changes into consideration is that we may wish to use different models for different sequential tasks, such as buying and selling shares in a stock market. This was the main reason for introducing gated Bayesian networks (GBNs) [4, 5], allowing the investor to create different BNs for the different phases of trading.[1] Another reason may be that the system that the modelled variables represent undergoes regime changes, i.e. there may be

[1] The GBN model also allows completely different random variables within each BN, something that we shall not explore further in this paper.

states of the world among which the independencies and distributions over the variables are different [6].

From the view of graphical models, the archetype approach for modelling regimes is to use a hidden Markov model (HMM), where the regimes are modelled using hidden random variables, and we observe the random variables that we are modelling under different states of these hidden variables. When using standard HMMs, it is common to assume that the observable variables are independent of each other given the hidden regime variable, and not to model any potential dependencies among the observed variables directly.

In this paper we are proposing an extension of the HMM, which we shall call GBN-HMM, where we bring in two of the fundamental ideas behind the GBN model. First, we shall allow for different BNs over the observable variables under the different states of the hidden variables, to have a regime-dependent model over the observable variables. Second, we shall model a potential dependence between one of the observable variables and the next hidden state. The second extension stems from one of the building blocks of GBNs, where the change of state is dependent on the posterior probability of a specific variable. The main difference between the GBN-HMM and the GBN is that GBNs identify one distinct BN as the model that represents the current regime, whereas the GBN-HMM defines a mixture of independence models, thus being a generative model of the data.

The rest of the paper is disposed as follows. In Section 2 we shall consider other existing extensions of the HMM, found in the literature, that are related to the extension that we shall propose. In Section 3 we will introduce and define the model that we are proposing, describing some of its underlying properties. Since there are hidden variables in the proposed model, parameter estimation is not immediately straightforward, and we shall therefore explore how we can use expectation maximisation (EM) in Section 4 to estimate the parameters of our model. In Section 5 we wish to demonstrate the appropriateness of the GBN-HMM using synthetic data, and compare it with the HMM as well as two other HMM variants. We then turn our attention to using the GBN-HMM in a real-world situation, namely trading shares in a stock market, in Section 6. Finally, we shall end this paper with our conclusions and a summary in Section 7.

2 RELATED WORK

HMMs have been applied and extended extensively throughout the literature, and we shall here not attempt an overview of all that has been explored. The interested reader may instead wish to consider the summary provided by Murphy [7]. Instead, we shall pay brief attention to a few existing variations that have a connection with the ideas that we are putting forward in this paper.

In [8] a HMM is described were some *control signal* is given as input to the hidden state and the observable variables, and offer an EM algorithm to update the parameters of the observational and transition distributions conditional on a sequence of input. As a variation on this theme, [9] proposes that transitions between hidden states in a HMM may not only depend on the immediately previous state, but also on the immediately proceeding observation. This potential dependence between the observed variables at time t and the hidden state at time $t+1$ is also present in the GBN-HMM that we are proposing. We shall use the model proposed in [9] as a comparison model in our experiments.

The auto-regressive HMM (AR-HMM), also know as the regime switching Markov model [10], incorporates potential dependence directly between an observable variable at time t and its counterpart at $t+1$. While the AR-HMM may be extended to higher orders, i.e. allowing for even longer dependence than only between t and $t+1$, the dependence is between counterparts in each time slice. However, dynamic Bayesian multinets (DBMs) proposed in [11] allow not only for dependence across time slices among observational counterparts, but arbitrarily among the observed variables. Furthermore, DBMs allow these potential dependencies to change depending on the hidden states, thus allowing for a more complex dependence structure across time. The model that we are proposing does not include potential direct dependence among observable variables across time, but rather within each time slice.

In the next section we shall formally introduce the GBN-HMM that we are proposing, and then subsequently discuss parameter estimation and experiments comparing the GBN-HMM with other HMM variants.

3 MODEL DEFINITION

The GBN-HMM that we are proposing consists of a set of discrete random variables $H_{1:T} = \{H_1, H_2, ..., H_T\}$ that represent the hidden state at each time $t \in [1, T]$. We call these the hidden state variables, and they each have the same number of possible states N. We use h_t to denote a specific instantiation of the variable H_t, and use $h_{j:k}$ to denote a sequence of states from time j to k. For each t, we will also model a set of discrete random variables $\mathbf{O}_t = \{O_t^1, O_t^2, ..., O_t^M\}$ for which we can observe their values. We will refer to these variables as the observable variables. We let $\mathbf{o}_t = \{o_t^1, o_t^2, ..., o_t^M\}$ be a particular instantiation of the observable variables at time t, and use $\mathbf{O}_{j:k}$ and $\mathbf{o}_{j:k}$ when considering all observable random variables and their respective values from time j to k.

Since we wish to model the observable variables depending on the current state, we will have one BN for each state of the hidden state variable H_t, that is, there are N BNs over the variables $\mathbf{O}_t$, and the value of H_t selects one of these. One of the variables in $\mathbf{O}_t$ is of particular interest, as we will model a potential dependence between this variable and the state of H_{t+1}. We will refer to this variable as the Z variable when we need to distinguish it from the other observable variables. Notation wise we let Z_t represent the Z variable at time t, and z_t an instantiation of the Z variable at time t.

Note that, although not made explicit, we have made use of certain independence assumptions among the variables $H_{1:T}$ and $\mathbf{O}_{1:T}$. First, we assume that $\mathbf{O}_t$ are conditionally independent of all previous random variables $\mathbf{O}_{1:t-1}$ and $H_{1:t-1}$, given the current hidden state variable H_t (thus knowing the current state renders the past irrelevant). Second, the current hidden state variable H_t is conditionally independent of $\mathbf{O}_{1:t-1} \setminus Z_{t-1}$ and $H_{1:t-2}$ given H_{t-1} and Z_{t-1} (thus knowing the value of the previous state and Z variable renders the rest of the past irrelevant). We can represent these assumptions using a graph, an example of which is depicted in Figure 1. In the figure we can see that it is O_t^3 that is the Z variable, as we are modelling a potential dependence between it and the next hidden state.

The final assumption that we will make is that of stationarity of the model. That is, the distributions and independencies that govern the model are independent of t. This implies that the probability of moving from one hidden state to another is the same regardless of t, and that the BNs selected by H_i are the same as for H_j for all $i, j \in [1, T]$. Furthermore, the Z variable is always the same observable variable, regardless of t or the state of H_t.

3.1 Factorisation

Using the independence assumptions implied by the model, and the chain rule of probability, we can factorise the joint distribution over $H_{1:T}$ and $\mathbf{O}_{1:T}$ into marginal and conditional distributions that together require fewer parameters than the full joint. To illustrate this factorisation in a succinct manner, we shall factorise the GBN-HMM given in Figure 1. We assume that the hidden state variables have two states, i.e. $N = 2$, however expanding this example to any number of observable variables, hidden states and time steps is straightforward. We begin the example by observing that we can isolate the variables O_3^1, O_3^2 and O_3^3 by conditioning on H_3 alone, which follows from Equation 2.

$$
\begin{aligned}
&p(O_1^1, O_1^2, O_1^3, \ldots, O_3^1, O_3^2, O_3^3, H_1, H_2, H_3) = \\
&p(O_3^1, O_3^2, O_3^3 | \cancel{O_1^1}, \cancel{O_1^2}, \cancel{O_1^3}, \ldots, \cancel{H_1}, \cancel{H_2}, H_3) \times \\
&p(O_1^1, O_1^2, O_1^3, \ldots, H_1, H_2, H_3) = \\
&p(O_3^1, O_3^2, O_3^3 | H_3) \times \\
&p(O_1^1, O_1^2, O_1^3, \ldots, H_1, H_2, H_3)
\end{aligned}
\tag{2}
$$

Since the hidden variables in a GBN-HMM select among several BNs over the observable variables, the two states of H_3 select between two joint distribution specifications over O_3^1, O_3^2 and O_3^3. If we let $pa_j(O_3^1)$ represent the parents of the variable O_3^1 with respect to the DAG of the BN selected by $H_3 = j$, then using the local Markov property of BNs we can continue the factorisation according to Equation 3.

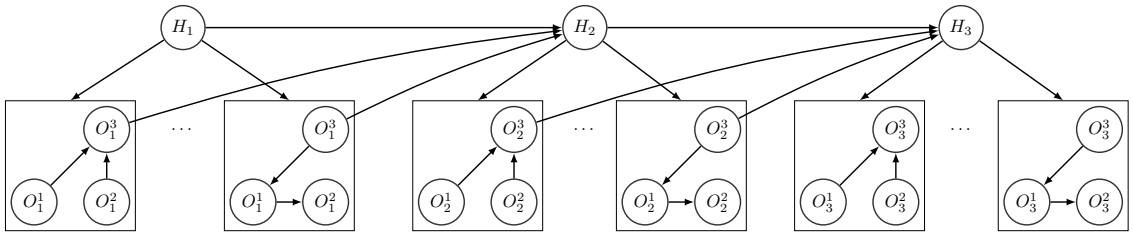

Figure 1: Graph representation of the GBN-HMM with three time steps.

$$
\begin{aligned}
&p(O_3^1, O_3^2, O_3^3|H_3)\times \\
&p(O_1^1, O_1^2, O_1^3, \ldots, H_1, H_2, H_3) = \\
&p(O_3^1, O_3^2, O_3^3)^{\delta(H_3=1)} p(O_3^1, O_3^2, O_3^3)^{\delta(H_3=2)}\times \\
&p(O_1^1, O_1^2, O_1^3, \ldots, H_1, H_2, H_3) = \\
&\prod_{j=1}^{2}\prod_{i=1}^{3} p(O_3^i|pa_j(O_3^i))^{\delta(H_3=j)}\times \\
&p(O_1^1, O_1^2, O_1^3, \ldots, H_1, H_2, H_3)
\end{aligned}
\tag{3}
$$

In Equation 3 we let $\delta(H_3 = j)$ represent the Kronecker delta, i.e. when H_3 takes the value j it equates to unity, otherwise zero.

The next step of the factorisation is to break out H_3 from the remaining variables, which follows from Equation 4. It should then be clear that we can continue the same operations for the remainder of the variables, ending the factorisation with a marginal distribution over H_1.

$$
\begin{aligned}
&\prod_{j=1}^{2}\prod_{i=1}^{3} p(O_3^i|pa_j(O_3^i))^{\delta(H_3=j)}\times \\
&p(H_3|\cancel{O_1^1}, \cancel{O_1^2}, \cancel{O_1^3}, \cancel{O_2^1}, \cancel{O_2^2}, O_2^3, \cancel{H_1}, H_2)\times \\
&p(O_1^1, O_1^2, O_1^3, O_2^1, O_2^2, O_2^3, H_1, H_2) = \\
&\prod_{j=1}^{2}\prod_{i=1}^{3} p(O_3^i|pa_j(O_3^i))^{\delta(H_3=j)} p(H_3|H_2, O_2^3)\times \\
&p(O_1^1, O_1^2, O_1^3, O_2^1, O_2^2, O_2^3, H_1, H_2)
\end{aligned}
\tag{4}
$$

The GBN-HMM factorisation for T time steps, with N hidden states and M observable variables, is given in Equation 5.

$$
\begin{aligned}
&p(H_1)\prod_{t=2}^{T} p(H_t|H_{t-1}, Z_{t-1})\times \\
&\prod_{t=1}^{T}\prod_{j=1}^{N}\prod_{i=1}^{M} p(O_t^i|pa_j(O_t^i))^{\delta(H_t=j)}
\end{aligned}
\tag{5}
$$

3.2 Likelihood

Considering a specific sequence of observations $\mathbf{o}_{1:T}$ and hidden states $h_{1:T}$, we can use the factorisation to compute the likelihood of this data under a set of parameters Θ. We let π_i represent the probability $p(H_1 = i|\Theta)$, a_{ijk} the probability $p(H_t = j|H_{t-1} = i, Z_{t-1} = k, \Theta)$, and $b_j^i(\mathbf{o}_t)$ represent the probability $p(O_t^i = o_t^i|pa_j(O_t^i) = \mathbf{o}_t^{pa_j(O_t^i)}, \Theta)^{\delta(H_t=j)}$, where we let $\mathbf{o}_t^{pa_j(O_t^i)}$ represent the values that the parent set takes in $\mathbf{o}_t$. Then the likelihood $p(\mathbf{o}_{1:T}, h_{1:T}|\Theta)$ can be expressed by Equation 6.

$$
\begin{aligned}
&p(\mathbf{o}_{1:T}, h_{1:T}|\Theta) = \\
&\pi_{h_1}\prod_{t=2}^{T} a_{h_{t-1},h_t,z_{t-1}}\prod_{t=1}^{T}\prod_{i=1}^{M} b_{h_t}^i(\mathbf{o}_t)
\end{aligned}
\tag{6}
$$

If we could observe both $\mathbf{o}_{1:T}$ and $h_{1:T}$ then estimating the parameters Θ that maximised the likelihood would be straightforward. However, since $H_{1:T}$ are hidden variables we cannot observe their values, and must therefore apply a more involved technique for estimating Θ.

4 PARAMETER ESTIMATION

The canonical way of solving the parameter estimation problem in regular HMMs (and in their ex-

tensions) is to employ EM. We shall also adopt this approach, and in this section describe the computations necessary for iteratively updating the parameters Θ for the GBN-HMM that we are currently proposing.

As before, let $\mathbf{o}_{1:T}$ represent a sequence of observations over the variables $\mathbf{O}_{1:T}$ and let $h_{1:T} = \{h_1, h_2, ..., h_T\}$ represent a sequence of states. Let $\mathcal{H}$ represent the set of all state sequences $h_{1:T}$. The current parameters for our model are denoted Θ', and we seek parameters Θ such that $p(\mathbf{o}_{1:T}|\Theta) \geq p(\mathbf{o}_{1:T}|\Theta')$. It can be shown [12] that this task can be converted into a maximisation problem of $Q(\Theta, \Theta') = \sum_{h_{1:T} \in \mathcal{H}} p(\mathbf{o}_{1:T}, h_{1:T}|\Theta') \log p(\mathbf{o}_{1:T}, h_{1:T}|\Theta)$.

Substituting $p(\mathbf{o}_{1:T}, h_{1:T}|\Theta)$ in the Q function with the likelihood expression in Equation 6, gives us the expanded Q function in Equation 7. From this expansion we can conclude that the individual terms do not interact, thus they can be maximised separately.

$$\begin{aligned} &Q(\Theta, \Theta') = \\ &\sum_{h_{1:T} \in \mathcal{H}} p(\mathbf{o}_{1:T}, h_{1:T}|\Theta') \log p(\mathbf{o}_{1:T}, h_{1:T}|\Theta) = \\ &\sum_{h_{1:T} \in \mathcal{H}} p(\mathbf{o}_{1:T}, h_{1:T}|\Theta') \log \pi_{h_1} + \\ &\sum_{h_{1:T} \in \mathcal{H}} p(\mathbf{o}_{1:T}, h_{1:T}|\Theta') \sum_{t=2}^{T} \log a_{h_{t-1}, h_t, z_{t-1}} + \\ &\sum_{h_{1:T} \in \mathcal{H}} p(\mathbf{o}_{1:T}, h_{1:T}|\Theta') \sum_{t=1}^{T} \sum_{i=1}^{M} \log b^i_{h_t}(\mathbf{o}_t) \end{aligned} \tag{7}$$

The derivation of which values for the individual terms that maximise the Q function is relatively lengthy. We therefore defer all details to the supplementary material[2], and here only account for the results of the derivation and show how to compute the necessary quantities.

4.1 Estimating new parameters

Computing new parameters π_i for the initial hidden state distribution that maximise the Q function is done according to Equation 8. Here we are taking the conditional probability of each possible state N given the observed data and the current parameters Θ'.

$$\pi_i = \frac{p(\mathbf{o}_{1:T}, h_1 = i|\Theta')}{p(\mathbf{o}_{1:T}|\Theta')} \tag{8}$$

The new parameters a_{ijk} can be computed using Equation 9, where we use $\delta(z_{t-1} = k)$ to represent the Kronecker delta which is unity when z_{t-1} takes on value k, and zero otherwise. Essentially, we are taking into consideration the expected number of times that we have observed a transition from state i to j when z took value k, divided by the expected number of times we have seen transitions away from i when z took value k.

$$a_{ijk} = \frac{\sum_{t=2}^{T} p(\mathbf{o}_{1:T}, h_{t-1} = i, h_t = j|\Theta')\delta(z_{t-1} = k)}{\sum_{t=2}^{T} p(\mathbf{o}_{1:T}, h_{t-1} = i|\Theta')\delta(z_{t-1} = k)} \tag{9}$$

The final set of parameters that we shall compute to maximise Q are the parameters of the distributions over the observed variables. We let b^i_{jkl} denote the parameter of the distribution for observable variable i when it takes on value l, given the hidden state j and its k:th parent configuration. An observation $\mathbf{o}_t$ will identify one such parameter for each observable variable under a specific hidden state. We let $\delta(\mathbf{o}_t, b^i_{jkl})$ represent the Kronecker delta such that it is unity when the parameter identified by $\mathbf{o}_t$ given $h_t = j$ is b^i_{jkl}, and zero otherwise, and likewise let $\delta(\mathbf{o}_t, b^i_{jk})$ be unity when the k:th parent set is identified given hidden state j (regardless of the value of l). We can then compute each b^i_{jkl} such that Q is maximised using Equation 10. This can again be seen as dividing the number of times that we expect to encounter a certain event (j, k, l) with the number of times we expect to encounter a superset of these events (j, k).

$$b^i_{jkl} = \frac{\sum_{t=1}^{T} p(\mathbf{o}_{1:T}, h_t = j|\Theta')\delta(\mathbf{o}_t, b^i_{jkl})}{\sum_{t=1}^{T} p(\mathbf{o}_{1:T}, h_t = j|\Theta')\delta(\mathbf{o}_t, b^i_{jk})} \tag{10}$$

4.2 Computing necessary quantities

While Equation 8, 9 and 10 describe which quantities are needed to compute the values necessary to maximise Q, the calculation of these quantities are not immediately available. In this section we turn

[2]Please find the supplementary material here: `https://www.ida.liu.se/~marbe92/pdf/gbn-hmm.supp.pdf`

our attention to the computation of these necessary quantities. As before, we defer some of the details to the supplementary material, and here offer the results from the derivation.

The two quantities that we require, which we shall call γ and ξ, are presented and expanded in Equation 11 and 12. Apart from the quantities α and β, the expansions consists of known quantities (readily available from the model under parameters Θ').

$$\begin{aligned}&\gamma_j(t) = p(\mathbf{o}_{1:T}, h_t = j|\Theta') = \\ &p(\mathbf{o}_{t+1:T}|\mathbf{o}_t, h_t = j, \Theta')p(\mathbf{o}_{1:t}, h_t = j|\Theta') = \\ &\beta_j(t)\alpha_j(t)\end{aligned} \tag{11}$$

$$\begin{aligned}&\xi_{ij}(t) = p(\mathbf{o}_{1:T}, h_{t-1} = i, h_t = j|\Theta') = \\ &p(\mathbf{o}_{t+1:T}|\mathbf{o}_t, h_t = j, \Theta')p(\mathbf{o_t}|h_t = j, \Theta')\times \\ &p(h_t = j|\mathbf{o}_{t-1}, h_{t-1} = i, \Theta')\times \\ &p(\mathbf{o}_{1:t-1}, h_{t-1} = i|\Theta') = \\ &\beta_j(t)\prod_{k=1}^{M} b_j^k(\mathbf{o}_t)a_{ijz_{t-1}}\alpha_i(t-1)\end{aligned} \tag{12}$$

What is left to do is to define recursively α and β, and then all required quantities are either already available or computable. We finish this section by defining these two quantities in Equation 13 and 14.

$$\begin{aligned}&\alpha_j(t) = p(\mathbf{o}_{1:t}, h_t = j|\Theta') = \\ &\prod_{k=1}^{M} b_j^k(\mathbf{o}_t)\sum_{i=1}^{N} a_{ijz_{t-1}}\alpha_i(t-1)\end{aligned} \tag{13}$$

$$\begin{aligned}&\beta_j(t) = p(\mathbf{o}_{t+1:T}|\mathbf{o}_t, h_t = j|\Theta') = \\ &\sum_{i=1}^{N}\beta_i(t+1)\prod_{k=1}^{M} b_i^k(\mathbf{o}_{t+1})a_{jiz_t}\end{aligned} \tag{14}$$

Note that the equations given here are slightly different from those used when estimating the parameters of a traditional HMM. In Equation 9 we are only considering cases under different values of the Z variable, and in Equation 10 we are considering different parent configurations rather than just the hidden state. Also, the definition of β in Equation 14 includes conditioning on $\mathbf{o}_t$, since the Z variable at time t may influence the hidden state at $t+1$.

The only part that is left to take into consideration is how we find the parent sets of each observable variable within each hidden state, i.e. how do we learn the structure of the BNs. We shall take this into consideration in the next section, and then move on to synthetic and real-world experiments.

4.3 Structure learning

Taking the approach of [13], we wish to identify the model over the observable variables that, together with the parameters, maximises the last term of Equation 7. While advances in exact learning of graphical model structures have been made [14, 15], we shall here rely on a heuristic approach. Therefore, we use a greedy thick thinning algorithm [16] to identify the structure over the observed variables, such that the term over the observable variables is maximised in Equation 7. Thus within each iteration of the EM algorithm, we also heuristically identify the best structure over the observed variables within each regime.

5 EXPERIMENTS USING SYNTHETIC DATA

We shall in this section account for our experiments using synthetic data to compare the GBN-HMM with three other models. The comparison models are: the standard HMM with observation variables that are independent of each other given the hidden state, the SDO-HMM proposed in [9], where observations are again independent given the hidden state, but where we have (using our term) a Z variable, and finally a version of our GBN-HMM but without the Z variable, which we shall call MULTI-HMM (due to their relationship to Bayesian multinets).

5.1 Methodology and data generation

A single sample was generated as follows (with input to the procedure the predictive power of the Z variable):

Four BNs were created by randomly generating four DAG structures[3] over four variables, and then

[3]We used the R package bnlearn which uses the method

uniformly at random generating parameters for the resulting conditional distributions.[4] The number of states for each variable was determined uniformly between two and five, except for the Z variable which was given four states.

The first data point in the sample was generated from the first BN. The value of the Z variable then determined which BN to take the second data point from, with a certain level of predictiveness (the supplied predictive power). For instance, if the Z variable took value two, and the predictive power was 0.6, then there was a 60% chance that the next data point would come from the second BN, and a 40% that the next data point would come from the same BN as the previous data point. We repeated this until there were 1000 data points in the sample.

Following this procedure we generated 50 samples for each of the predictive powers 0.6, 0.7, 0.8 and 0.9.

For the synthetic experiments we were interested in how well the models fit held out test data. Therefore, for each sample, we employed a 5-fold cross-validation procedure using two thirds of the data to determine the number of hidden states, estimate the parameters of the models, and to learn the BN structures for GBN-HMM and MULTI-HMM. For SDO-HMM and GBN-HMM the models were told which Z variable to use. The remaining third was treated as held out test data, the likelihood of which will be reported.

5.2 Results and discussion

In Table 1 the results from the synthetic experiments are reported. Each row represents a certain predictive power. The values in the table are the means of the log-likelihoods of the held out test data, over the 50 samples, given each model.

Already when the Z variable has a predictive power of 0.6, the GBN-HMM had a considerably better fit to the data than both HMM and MULTI-HMM (note that this is log-scale). However, the SDO-HMM was also able to utilise this predictive power to get a similar fit as the GBN-HMM. As the predictive power of the Z variable increased to 0.7, the difference between the GBN-HMMs' fit of the data and the other models increased, suggesting that taking this predictiveness into account,

Table 1: Means of log-likelihoods of held out data, using different predictive powers of the Z variable.

HMM	SDO-HMM	MULTI-HMM	GBN-HMM
Predictive power = 0.6			
-1546.438	-1537.453	-1558.732	**-1536.529**
Predictive power = 0.7			
-1538.541	-1526.940	-1550.820	**-1518.590**
Predictive power = 0.8			
-1535.269	-1509.830	-1546.112	**-1506.726**
Predictive power = 0.9			
-1513.058	-1476.529	-1526.843	**-1475.436**

and allowing for multiple BNs, can improve the appropriateness. When we look at the outcomes when the predictive power was increased to 0.8 and 0.9, the two models that do not utilise a Z variable (HMM and MULTI-HMM) drift further from the GBN-HMM, while the HMM-SDO reversed and came closer again. Although the GBN-HMM outperforms the other models throughout all experiments, it is interesting to see that the SDO-HMM can outperform HMM and MULTI-HMM by utilising the Z variables predictive power.

While the experiments that we have reported in this section work well as a confirmation of the proposed model's appropriateness, we shall now turn our attention to experiments where we wish to employ the model for a specific task. In Section 6 we shall explore the performance of the four models when they are used for systematic stock market trading.

6 TRADING THE STOCK MARKET

In this section we shall employ the models under comparison for trading stock shares, with the goal of balancing the risk and reward of such trading.

We shall first offer a brief introduction to some of the ideas and concepts surrounding systematic stock trading, and then employ the GBN-HMM in such trading, using the same models as in Section 5 as comparison (HMM, SDO-HMM and MULTI-HMM).

proposed in [17] to generate DAGs uniformly at random.

[4] Using the method described in [18].

6.1 Systematic stock trading concepts

The general idea of systematic stock trading is to use some collected data to create rules that identify opportune times to own certain stock shares, and times when it is less beneficial to own them. Usually this is referred to as generating buy and sell *signals*. For the purpose of the experiments that we shall undertake, this type of all-or-nothing approach will suffice. However, in a more mature systematic trading system one may very well wish to trade several different shares at different quantities, utilising diversification in one's favour.

If signals from a system are executed, then this will generate a certain risk and reward in terms of the initial investment. For instance, if we execute a buy signal then any change in the price of the bought shares will also give us a proportional (positive or negative) return on our investment. Naturally, one seeks a positive return on one's investment, however simply using the raw return as the only goal of investment is not necessarily the best approach. Instead it is common to take into consideration the variation of the returns an investment yields. Therefore we shall seek a high Sharpe ratio (named after Nobel Laureate William F. Sharpe), where we take the mean of our returns, less the *risk free rate*, divided by the standard deviation of our returns. Here, the risk free rate is the return that we can expect from interest, or some other "safe" asset such as government bonds. As our comparison will be among models, rather than investment strategies, we shall remove the risk free rate from the Sharpe ratio and simply consider the mean return divided by the standard deviation of the returns.

The type of data that is used in stock trading systems vary greatly, however a common approach is to take the historical price and apply so called *technical analysis indicators* to gauge whether prices are trending, shares are overpriced, etc. For our purposes we shall consider two such indicators: the relative difference between two *moving averages*, often referred to as MACD [19], and the relative strength index (RSI) [20], which compares recent price increases with recent price decreases. The MACD is computed by first calculating two moving averages with different length windows, one using the most recent five days of prices, and one using the most recent ten days of prices. The difference between the two then becomes a gauge for the trend in the market, if it is positive it means that the five day moving average is above the ten day moving average, indicating an upswing in price (and vice versa). The RSI computes the average of all price increases the past 14 days and divides by the average of all price decreases the past 14 days, a high RSI indicates that prices have been increasing strongly and may therefore be overpriced (and vice versa). For sake of brevity we shall leave out the exact calculations of these indicators, and refer the interest reader to the referred literature.

6.2 Methodology

The MACD and RSI gave us two observable variables in our models, and we additionally considered the first order backward difference of these variables (i.e. we approximated the indicators' first order derivatives), giving us a total of four observed variables. The MACD was discretised into two states, positive and negative, and used as the Z variable. The rest of the indicators were discretised into four states, using their respective mean and one standard deviation below and above their mean as cut points.

We used daily data between 2003-01-01 and 2012-12-28 for seven actively traded stocks: Apple (AAPL), Amazon (AMZN), IBM (IBM), Microsoft (MSFT), Red Hat (RHT), Nvidia (NVDA) and General Electric (GE). To create multiple simulations from this data we divided the data into ten blocks (one year per block), and created seven simulations by first using block one, two and three as training data and block four as testing data, and then block two, three and four as training data and block five as testing data, and so on.

As in the experiments in Section 5, we employed a 5-fold cross-validation procedure using the training data to decide upon the number of hidden states, the parameters of the models, and the BN structures within the GBN-HMMs and MULTI-HMMs. For SDO-HMM and GBN-HMM the models were told to use the MACD variable as the Z variable.

While ones first intuition may be to attempt to label the hidden states of our models as "buy", "sell", etc. and thereby generate signals that can be executed, this is not the approach we will take

in this application. We do not know how many hidden states will be identified in each simulation, thus it would require some automatic labelling based on the number of states and historical advantage of different types of labelling. Instead, we shall build our rules as follows:

- On day t, when we know the values of $\mathbf{O}_{1:t}$, we shall make a prediction of the MACD variable at time $t+1$.
- If $\mathrm{p}(MACD_{t+1} = \text{positive} \mid \mathbf{O}_{1:t}) > \theta$, then generate a buy signal.
- If $\mathrm{p}(MACD_{t+1} = \text{negative} \mid \mathbf{O}_{1:t}) > \theta$, then generate a sell signal.

We are thus generating buy and sell signals when enough of the probability mass indicates that the MACD is positive/negative. The particular θ used was determined for each model by generating trade signals using the training data. For each simulation we generated signals for each block reserved for training (three blocks per simulation) and calculated the Sharpe ratio per block using different θ (0.50, 0.55, ..., 0.90, 0.95). The θ used on the test data was then the θ with the highest average Sharpe ratio over the training blocks.

6.3 Results and discussion

Signals were generated for each held out test block, and the annual return and standard deviation was calculated for each block and model, giving rise to an annual Sharpe ratio for each model and traded stock. The annual Sharpe ratios are given in Table 2.

From the table we can see that the use of multiple BNs (i.e. MULTI-HMM and GBN-HMM) yields a higher annual Sharpe ratio for five of the seven stocks, losing out to SDO-HMM for RHT and HMM for IBM. In four out of the five cases where using multiple BNs was better, the GBN-HMM outperformed the MULTI-HMM. Thus in general, allowing for multiple BNs over the observable variables does increase the performance of the trading systems. Similarly, when considering the models that include a Z variable against those which did not, we see that the Z variable models won five against two. When comparing the use of both multiple BNs and a Z variable, i.e. the GBN-HMM, the outcome is four against three in favour of the GBN-HMM. So even when all the other models are counted as one, the GBN-HMM wins. It should be noted that the models are all generative, thus they have been learnt with the goal of explaining the data generating process, and not to the specific task of stock trading. The case of SDO-HMM outperforming GBN-HMM on RHT is evidence of this difference between goals, as the GBN-HMM should always explain the data better, or the same, as the SDO-HMM, as the former is capable of mimicking the same structure as the latter.

Table 2: Annual Sharpe ratio comparison.

HMM	SDO-HMM	MULTI-HMM	GBN-HMM
Apple (AAPL)			
0.844	0.708	**0.849**	0.718
Amazon (AMZN)			
0.466	0.580	0.449	**0.592**
IBM (IBM)			
0.713	0.521	0.699	0.616
Microsoft (MSFT)			
0.091	-0.189	-0.307	**0.219**
Red Hat (RHT)			
-0.198	**0.111**	-0.780	-0.085
Nvidia (NVDA)			
0.113	0.211	0.262	**0.308**
General Electric (GE)			
0.0621	0.362	-0.378	**0.419**

It seems that the different models are advantageous under different circumstance, although the GBN-HMM seems to have an advantage in general. However, since GBN-HMMs embrace the other three models, we could take the structure learning further than only for the individual BNs, and learn which one of the four models considered is the most appropriate for the current task. We however leave such exploration to future work.

7 CONCLUSIONS & SUMMARY

Many real-world systems undergo changes over time, perhaps due to human intervention or natural

causes, and we do not expect probabilistic relationships among the random variables that we observe to stay static throughout these changes. We therefore find the use of multiple BNs for the different resulting regimes intriguing. In this paper we have proposed a model, which we call GBN-HMM, that incorporates these regime changes by using different BNs for the different regimes. Furthermore, the GBN-HMM allows us to model potential driving forces behind the regime changes by utilising some observational variables. We have shown the benefits of using the GBN-HMM in comparison with three related models, both by comparing fitness to data using synthetic data, and in a real-world systematic trading task.

References

[1] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers, 1988.

[2] F. V. Jensen and T. D. Nielsen, *Bayesian networks and decision graphs*. Springer, 2007.

[3] K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*. Taylor and Francis Group, 2011.

[4] M. Bendtsen and J. M. Peña, "Gated Bayesian networks for algorithmic trading," *International Journal of Approximate Reasoning*, vol. 69, pp. 58–80, 2016.

[5] M. Bendtsen, "Bayesian optimisation of gated Bayesian networks for algorithmic trading," in *Proceedings of the Twelfth Annual Bayesian Modeling Applications Workshop*, pp. 2–11, 2015.

[6] M. Bendtsen, "Regimes in baseball player's career data," *Data Mining and Knowledge Discovery*, 2017, accepted.

[7] K. P. Murphy, *Machine learning: a probabilistic perspective*. The MIT press, 2012.

[8] S. Bengio and Y. Bengio, "An EM algorithm for asynchronous input/output hidden Markov models," in *Proceedings of the International Conference On Neural Information Processing*, pp. 328–334, 1996.

[9] Y. Li, "Hidden Markov models with states depending on observations," *Pattern Recognition Letters*, vol. 26, no. 7, pp. 977–984, 2005.

[10] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989.

[11] J. A. Bilmes, "Dynamic Bayesian multinets," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 38–45, 2000.

[12] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2013.

[13] N. Friedman, "The Bayesian structural EM algorithm," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 129–138, 1998.

[14] D. Sonntag, J. M. Peña, A. Hyttinen, and M. Järvisalo, "Learning optimal chain graphs with answer set programming," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 822–831, 2015.

[15] C. Yuan and B. Malone, "Learning optimal Bayesian networks: a shortest path perspective," *Journal of Artificial Intelligence Research*, vol. 48, no. 1, pp. 23–65, 2013.

[16] D. Heckerman, "A tutorial on learning with Bayesian networks," Tech. Rep. MSR-TR-95-06, Microsoft Research, March 1995.

[17] G. Melançon, I. Dutour, and M. Bousquet-Mélou, "Random generation of directed acyclic graphs," *Electronic Notes in Discrete Mathematics*, vol. 10, pp. 202–207, 2001.

[18] J. S. Ide and F. G. Cozman, "Random generation of Bayesian networks," in *Brazilian Symposium on Artificial Intelligence*, pp. 366–376, 2002.

[19] J. J. Murphy, *Technical analysis of the financial markets*. New York Institute of Finance, 1999.

[20] W. J. Wilder, *New concepts in technical trading systems*. Trend Research, 1978.