# Regime Aware Learning

**Marcus Bendtsen** MARCUS.BENDTSEN@LIU.SE
*Department of Computer and Information Science*
*Linköping University (Sweden)*

## Abstract

We propose a regime aware learning algorithm to learn a sequence of Bayesian networks (BNs) that model a system that undergoes *regime* changes. The last BN in the sequence represents the system's current regime, and should be used for BN inference. To explore the feasibility of the algorithm, we create baseline tests against learning a singe BN, and show that our proposed algorithm outperforms the single BN approach. We also apply the learning algorithm on real world data from the financial domain, where it is evident that the algorithm is able to produce BNs that have adapted to the regime changes during the most recent global financial crisis of 2007-08.

**Keywords:** Regime changes; Bayesian networks; financial application.

## 1. Introduction

Many tasks performed by practitioners involve observing a system over time, and then making predictions and decisions based on the observed data. As long as the probabilistic relationships and distributions of the variables that make up the system stay the same, we can define a joint probability distribution over the variables and estimate the parameters from the data we have observed. For instance, a medical practitioner may monitor a patient's vital signs and create a model for the patient's overall health. However, for tasks such as spam detection, crime prediction, financial planning, as well as patient monitoring, it may be incorrect to assume that the probabilistic relationships and distributions among the modelled variables are stable. Estimating a single joint probability distribution over a set of variables that undergoes changes may lead to poor results.

The phenomenon of *regime* changes has been studied extensively in the fields of ecology (Scheffer et al., 2001; Andersen et al., 2009), economy and finance (Hamilton, 1989; May et al., 2008), and biology (Pal et al., 2013). Given the diversity of fields studying regimes, there is no general consensus of what a regime entails, nor its length or the abruptness of a regime change. However, a non-conflicting definition is to say that a regime is a *steady state* of some system under observation. We define this steady state as one where all probabilistic relationships and distributions of the variables in a system stay the same. The system can exhibit several regimes, between which the probabilistic relationships and distributions may change.

We also consider it possible for a system to return to a previous regime, thereby allowing regimes to reoccur. If no regimes reoccur, then a system's *regime transition structure* is a chain of regimes: $R_1 \to R_2 \to \cdots \to R_k$. However, if a system does exhibit regimes that reoccur, then the regime transition structure will contain cycles, e.g. $R_1 \to R_2 \rightleftarrows R_3 \to \cdots \to R_k$. We have previously shown how these regime transition structures can accurately be recovered from batch data (Bendtsen and Peña, 2016c,b), thereby gaining important insight into the individual regimes of a system, as well as how these regimes transition into one another. While the aim of our previous work was to uncover these regime transition structures in a given dataset, the aim of this paper is to discover the best model for the regime a system currently is in, and then update this model in a regime aware fashion each time a new data point is made available. The model that we will use to represent a system that undergoes regimes is a sequence of Bayesian networks (BNs), where the last BN in the sequence represents the current regime.

### 1.1 Bayesian Networks

Introduced by Pearl (1988), BNs consists of two major components: a qualitative representation of independencies among 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. 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 are defined with respect to the DAG of the BN. Let $\mathbf{X}$ be a set of random variables in a BN, and let $\Pi(X_i)$ be 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.

$$p(\mathbf{X}) = \prod_{X_i \in \mathbf{X}} p(X_i|\Pi(X_i)) \tag{1}$$

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 Pearl (1988); Korb and Nicholson (2011); Jensen and Nielsen (2007).

While a BN has advantages when representing a single independence model, it lacks the ability to represent several independence models simultaneously, i.e. it lacks the ability to model several regimes. Therefore, the learning algorithm that we will propose will generate a sequence of BNs, each one representing a regime of the system under observation, where the last BN represents the current regime and should therefore be used for BN inference.

### 1.2 Related Work

Refining or updating the structure and conditional distributions of a BN in response to new data has been studied for some time (Buntine, 1991; Lam and Bacchus, 1994; Friedman and Goldszmidt, 1997; Lam, 1998). However, these approaches assume that data is received from a stationary distribution, i.e. a system that does not undergo regime changes.

Nielsen and Nielsen (2008) approach the problem of having a stream of observations which they say is *piecewise stationary*, i.e. observations within a section of the stream come from the same distribution, but changes may occur into a new stationary section. Their goal is to incrementally learn the structure of a BN, adapting the structure as new observations are made available. They achieve this by monitoring local changes among the variables in the current network, and when a conflict occurs between what is currently learnt and what is observed, they refine the structure of the BN. Focusing only on local changes allows them to reuse all previous observations for parts of the BN that have not changed.

We do not make the assumption of local change between regimes, but allow for the structure of a BN to change arbitrarily between regimes. As we shall see in Section 2, this allows us to plug in any

BN structure learning algorithm into our proposed algorithm. Furthermore, we base the decision of identifying a regime shift on the posterior of the entire model, rather than a conflict in one variable.

Other approaches to adaptation exists, including using latent variables to model the changes explicitly (Borchani et al., 2015), and placing a probability distribution over the location of the most recent drift point (Bach and Maloof, 2010). We refer the interested reader to a survey on *concept drift* adaptation by Gama et al. (2014).

### 1.3 Outline

The rest of the paper is structured as follows. In Section 2 we will introduce the proposed regime aware learning algorithm. In Section 3 we will run a set of baseline experiments on synthetic data to show the feasibility of the proposed algorithm. In Section 4 we will show how the algorithm performs on real world data from the financial domain. Finally, in Section 5 we will summarise our current work and give some final remarks.

## 2. Learning Algorithm

In this section we will describe the proposed learning algorithm. The aim is to supply a model to the investigator, and to update this model each time there is new data available. We begin by accounting for the steps that the algorithm takes each time new data is available, and then explain the details of each step in the subsequent sections.

Let $\mathcal{D}$ represent an ordered dataset, and let a hypothesis $H$ be a division of $\mathcal{D} = \{d_1, ..., d_n\}$ into subsets. Let $\mathcal{A}$ represent an algorithm for learning the structure and parameters of a BN, and let $\mathcal{A}(\{d_l\})$ mean that $\mathcal{A}$ has been used to learn a BN using the data point $d_l$. A model $M$ is then created by using $\mathcal{A}$ to learn a BN for each subset defined by a hypothesis $H$. For instance, given five data points, the hypothesis $H\{2, 4\}$ splits the data at position 2 and 4, resulting in a model $M$ with the three BNs $\mathcal{A}(\{d_1\})$, $\mathcal{A}(\{d_2, d_3\})$ and $\mathcal{A}(\{d_4, d_5\})$. Notation wise, we will always let a hypothesis with a specific subscript define a model with the same subscript, i.e. the hypothesis $H_k$ defines the model $M_k$.

Each iteration of the algorithm will result in one hypothesis that defines the current model, we denote the current hypothesis by $H_C$ and the current model by $M_C$. The learning algorithm consists of the following five steps, which are run each time a new data point $d_i$ is made available. Before the first iteration, set $H_C = \emptyset$ and $\mathcal{D} = \emptyset$.

1. Add $d_i$ to the end of $\mathcal{D}$.

2. Create a set of hypotheses $\mathbf{H}$ by calling $\texttt{PROPOSE}(H_C, \mathcal{D})$.

3. Find $H_{max}$ such that $H_{max} = \arg\max_{H_j \in \mathbf{H}} \; p(M_j|\mathcal{D})$.

4. Set $H_C = H_{max}$ if $p(M_{max}|\mathcal{D}) > p(M_C|\mathcal{D})$.

5. Return to the investigator $M_C = \texttt{MERGE}(H_C, \mathcal{D})$. The BN in $M_C$ which was learnt using the subset of $\mathcal{D}$ containing $d_i$ represents the current regime.

The algorithm creates a set of hypotheses by calling PROPOSE, and then finds the hypothesis $H_{max}$ that defines the model $M_{max}$ with the highest posterior probability $p(M_{max}|\mathcal{D})$. If $p(M_{max}|\mathcal{D}) > p(M_C|\mathcal{D})$, i.e. if the posterior odds is greater than one in favour of the hypothesised

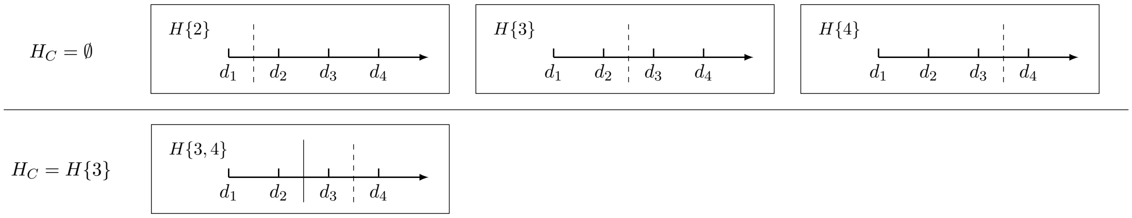


Figure 1: Effect of constraint on hypothesis generation

model, then $H_C$ is replaced by $H_{max}$. The MERGE procedure then exhaustively combines the subsets defined by $H_C$, in order to identify reoccurring regimes, and the final model $M_C$ is returned to the investigator. In the following sections we will begin by explaining the procedure PROPOSE, followed by how the posterior $p(M|\mathcal{D})$ is calculated, and finally the MERGE procedure. We note here that the algorithm does not assume that any splits exists in the data, as the initial $H_C$ is the empty set, and splits are only added to $H_C$ if the posterior odds are in favour of such an addition.

## 2.1 Proposing Hypotheses

In step 2 of the algorithm, PROPOSE creates a set of hypotheses $\mathbf{H}$, which becomes a search space in step 3. If every possible hypothesis was to be proposed, then the search space would expand beyond computational feasibility as $\mathcal{D}$ grows. We therefore introduce the following constraint, and give two examples that highlight where this constraint plays a role in the procedure:

Each hypothesis in $\mathbf{H}$ has the splits in $H_C$, plus a new one that is after the last split in $H_C$.

Figure 1 depicts two different outcomes of using PROPOSE when a dataset containing four points is available. The top example in the figure depicts the case where $H_C$ contains no splits. This situation has been reached by rejection of all hypotheses when $d_1$, $d_2$ and $d_3$ were collected. In this case there are three possible hypotheses, illustrated by dashed lines in the figure. Note that due to the constraint we cannot add two splits, e.g. the hypothesis $H\{2,3\}$ is not generated since it has two more splits than $H_C$. In the bottom example in the figure, $H_C$ contains one split at data point three (illustrated by a solid line). The only hypothesis generated in this case is $H\{3,4\}$, since due to the constraint, we cannot propose splits before the existing split at data point three.

## 2.2 Posterior of a Model

As we have seen from the discussion regarding the PROPOSE procedure, the number of hypotheses that are proposed is $n - d_{last}$, where $d_{last}$ is the last split in $H_C$ (or 1 if $H_C = \phi$). Therefore we must be able to choose one of these, and then compare this hypothesis with the current hypothesis in order to decide if it should be replaced. We do this by finding the hypothesis with the highest posterior $p(M|\mathcal{D}) \propto p(\mathcal{D}|M)p(M)$.

Let $H$ be a hypothesis with $k > 0$ splits $\delta_1, ..., \delta_k$, then the model $M$ that $H$ defines consists of $k+1$ BNs. Also let $\mathcal{D}$ contain $n$ data points, and let $\mathcal{D}_l^j$ represent the subset $\{d_l, ..., d_j\}$ $(l \leq j)$. We calculate the marginal likelihood of the data $\mathcal{D}$ given the model $M$ by the product of the marginal likelihoods of its $k+1$ BNs. When $k > 1$ we therefore have:

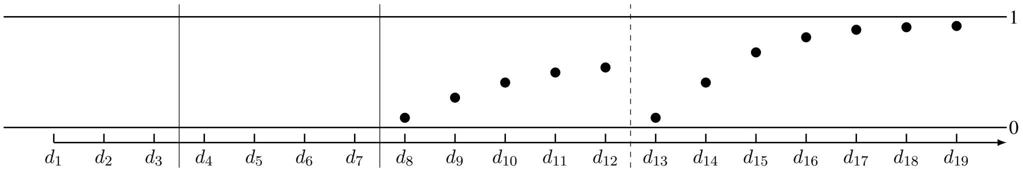


Figure 2: Stylistic view of the probabilities of the left and right subset sizes

$$p(\mathcal{D}|M) = p(\mathcal{D}_1^{\delta_1-1}|\mathcal{A}(\mathcal{D}_1^{\delta_1-1}))p(\mathcal{D}_{\delta_k}^n|\mathcal{A}(\mathcal{D}_{\delta_k}^n))\prod_{i=1}^{k-1} p(\mathcal{D}_{\delta_i}^{\delta_{i+1}-1}|\mathcal{A}(\mathcal{D}_{\delta_i}^{\delta_{i+1}-1})) \tag{2}$$

When $k = 1$ only the first two factors of Equation 2 apply, and when $k = 0$ we have $p(\mathcal{D}|M) = p(\mathcal{D}|\mathcal{A}(\mathcal{D}))$. Notice that the first $k - 1$ BNs are equal among all hypotheses (since they share the splits in $H_C$), thus the marginal likelihoods for these BNs need only to be calculated once, and can be reused in all following iterations.

The PROPOSE procedure will generate hypotheses where some or all of the subsets contain very few data points. There is therefore a risk that single values that are considered extreme given the current model will suggest that the current model be replaced. To avoid situations where many single point regimes are identified, we define a prior $p(M)$ over the models induced by the hypotheses, such that small regimes are less probable. We first conclude that all hypotheses share the splits from $H_C$, thus the prior over this part of the model is equal among hypotheses. The hypotheses do however differ with respect to the two new subsets that are created on either side of the added split, one to the *left* and one to the *right*. If a hypothesis was to replace $H_C$, then the subset to the left of the added split would be locked in as a regime, however the right subset will continue to increase in size as new data points become available. We therefore define $p(M)$ such that the probability of a specific subset size on the left side is smaller than the equivalent on the right side.

Let $G_{cdf}$ represent the cumulative distribution of a geometric distribution, and let $G_{cdf}(m, q)$ represent the probability of a regime with $m$ data points, when the geometric distribution is parameterised with $0 < q \leq 1$. We then define $p(M)$ by Equation 3, where $p(M_C)$ is equal among all hypotheses and therefore set to unity, and $G_{cdf}$ is taken to the $n$:th power to scale against the unnormalised $p(\mathcal{D}|M)$.

$$p(M) = p(M_C)G_{cdf}(\delta_k - \delta_{k-1}, q_{(left)})^n G_{cdf}(n - \delta_k + 1, q_{(right)})^n \tag{3}$$

We offer a stylistic view of $p(M)$ in Figure 2. In the figure, $H_C = H\{4, 8\}$, and the proposed hypothesis is $H\{4, 8, 13\}$. The black dot over $d_{12}$ indicates the probability of a regime with five data points ($\mathcal{D}_8^{12}$) given a geometric distribution parameterised with $q_{(left)}$, while the black dot over $d_{19}$ indicates the probability of a regime with seven data points ($\mathcal{D}_{13}^{19}$) given a geometric distribution parameterised with $q_{(right)}$. As we can see, the probabilities grow as the size of the subsets increase, and $q_{(left)} < q_{(right)}$ following the previous discussion.

### 2.3 Merging Subsets

The final step of the algorithm is to return the model implied by $H_C$ to the investigator. However, before doing so we perform one more task to improve upon the contained BNs. Naïvely assuming

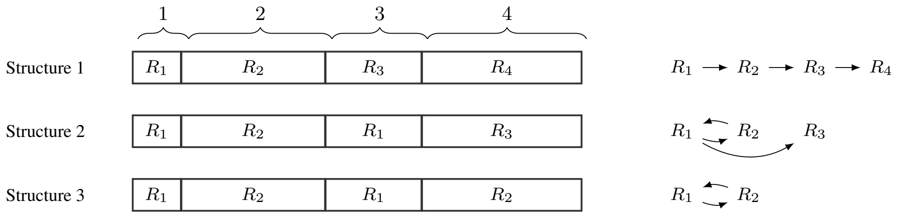


Figure 3: Example of merging subsets

that each split identifies a new regime would lead to a chain of regimes, i.e. if there were two splits in $H_C$ we would have $R_1 \to R_2 \to R_3$. We would then return this model to the investigator and tell them to use the BN representing $R_3$ as a model for the current regime. While it may be entirely possible for a system to exhibit this type of regime structure, it is also possible that $R_2$ transitioned back to $R_1$, and not into a new regime $R_3$. Therefore we must try each possible recursive merging of nonadjacent subsets, as defined by the splits in $H_C$, and score a new model based on these new subsets. Next follows an explanation and example of this merging.

In the example in Figure 3 we have identified three splits, resulting in four subsets of the data (labeled 1, 2, 3 and 4). The first possible structure requires no merging at all; it suggests that each subset identifies a new regime (depicted to the left) and the regime transition structure is therefore a chain of four regimes (depicted to the right). From the first structure we cannot merge subsets 1 and 2, since they are adjacent and we would not have a split here if the two subsets belonged to the same regime. A new structure can however be constructed by merging subsets 1 and 3, resulting in the second structure in the figure. Note that we have now labelled subset 3 with $R_1$ and subset 4 with $R_3$, as we now only have three regimes rather than four in the previous structure. The implied regime transition structure now contains cycles. From the second structure the only merging that can be done is to merge subsets 2 and 4, resulting in the third structure. Note that the example is not complete, as we should go back to the first structure and start the recursive procedure again, but this time by merging subsets 1 and 4 (and similarly so for 2 and 4).

In order to choose which structure that should be returned from one iteration of the proposed algorithm, we start with the chain of regimes that is given directly from the splits, and then continue to score each possible recursive merging. The merged subsets are scored using a simple rephrasing of Equation 2, and the structure giving the highest score represents the final model returned to the investigator.

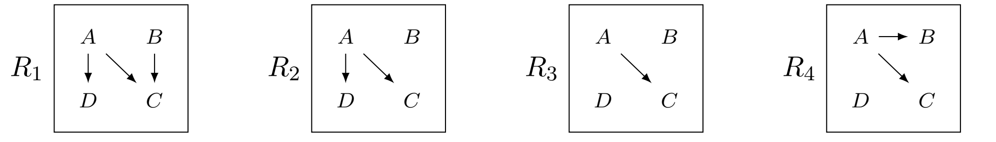
## 3. Experiments

In order to investigate the benefit of using the proposed learning algorithm, we set up a series of experiments to create a baseline comparison against learning a single BN. We considered datasets containing between zero and three regimes shifts. Since regimes may reoccur, a dataset containing more than one shift may be a sample from one of several regime transition structures. When considering datasets with $i$ shifts, coming from the $j$:th possible transition structure, we will denote such a pair with $\mathcal{S}_{i,j}$. Please see Table 1 for all pairs used throughout the experiments.

We ran two separate experiments, which we will refer to as exp-$c$ ($c$ for *calm*) and exp-$v$ ($v$ for *volatile*). For each experiment we set up a set of four BNs to represent the regimes. In the first set of BNs (set-$c$), we only added or removed one edge between regimes, thus there is still similarity

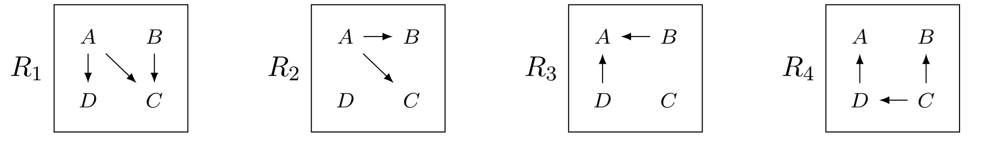
| Pair | Shifts | Structure | Pair | Shifts | Structure |
|---|---|---|---|---|---|
| $\mathcal{S}_{0,1}$ | 0 | $R_1$ | $\mathcal{S}_{3,1}$ | 3 | $R_1 \rightarrow R_2 \rightarrow R_3 \rightarrow R_4$ |
| $\mathcal{S}_{1,1}$ | 1 | $R_1 \rightarrow R_2$ | $\mathcal{S}_{3,2}$ | 3 | $\begin{tikzcd} R_1 \arrow[r, shift right] \arrow[rr, bend right] & R_2 \arrow[l, shift right] & R_3 \end{tikzcd}$ |
| $\mathcal{S}_{2,1}$ | 2 | $R_1 \rightarrow R_2 \rightarrow R_3$ | $\mathcal{S}_{3,3}$ | 3 | $\begin{tikzcd} R_1 \arrow[r] & R_2 \arrow[r, shift right] & R_3 \arrow[l, shift right] \end{tikzcd}$ |
| $\mathcal{S}_{2,2}$ | 2 | $R_1 \leftrightarrows R_2$ | $\mathcal{S}_{3,4}$ | 3 | $\begin{tikzcd} R_1 \arrow[r] & R_2 \arrow[r] & R_3 \arrow[ll, bend left] \end{tikzcd}$ |
| | | | $\mathcal{S}_{3,5}$ | 3 | $R_1 \leftrightarrows R_2$ |

Table 1: The number of shifts and transition structures under consideration



| $R_1$ | $R_2$ | $R_3$ | $R_4$ |
|---|---|---|---|
| $A \sim N(1, 0.5)$ | $A \sim N(1, 0.5)$ | $A \sim N(1, 0.5)$ | $A \sim N(1, 0.5)$ |
| $B \sim N(-1, 0.5)$ | $B \sim N(-1, 0.5)$ | $B \sim N(-1, 0.5)$ | $B \sim N(2A^2, 0.1)$ |
| $C \sim N(sin(2A + 2B), 0.1)$ | $C \sim N(sin(2A), 0.1)$ | $C \sim N(sin(2A), 0.1)$ | $C \sim N(sin(2A), 0.1)$ |
| $D \sim N(2A^2, 0.1)$ | $D \sim N(2A^2, 0.1)$ | $D \sim N(-2, 0.5)$ | $D \sim N(-2, 0.5)$ |

Figure 4: BNs in set-$c$



| $R_1$ | $R_2$ | $R_3$ | $R_4$ |
|---|---|---|---|
| $A \sim N(1, 0.5)$ | $A \sim N(1, 0.5)$ | $A \sim N(sin(2B + 2D), 0.1)$ | $A \sim N(sin(2D), 0.1)$ |
| $B \sim N(-1, 0.5)$ | $B \sim N(2A^2, 0.5)$ | $B \sim N(-1, 0.5)$ | $B \sim N(C^2, 0.1)$ |
| $C \sim N(sin(2A + 2B), 0.1)$ | $C \sim N(sin(2A), 0.1)$ | $C \sim N(2, 0.5)$ | $C \sim N(2, 0.5)$ |
| $D \sim N(2A^2, 0.1)$ | $D \sim N(-2, 0.5)$ | $D \sim N(-2, 0.5)$ | $D \sim N(sin(C^2), 0.1)$ |

Figure 5: BNs in set-$v$

between the BNs that make up the regimes. The BNs in set-$c$ are depicted in Figure 4, along with the individual variables' distributions. In the second set of BNs (set-$v$), we used the four BNs depicted in Figure 5, where there are more edge removals and additions between BNs, thus the similarity between regimes is drastically reduced.

### 3.1 BN Structure Learning and Priors

For all experiments we used a greedy thick thinning algorithm (Heckerman, 1995) for learning the structures of the individual BNs, where the marginal likelihood was the target to improve, thus this is a Bayesian approach that entails regularisation of the structure. Since we were dealing with continuous data with non-linear relationships between variables, we used Gaussian process priors for variables with parents during the learning (using the radial basis kernel). Variables without parents were assigned a normal-inverse-gamma prior.

The prior $p(M)$ in Equation 3 was parameterised with $q_{(left)} = 0.05$ and $q_{(right)} = 0.5$, thus it is less probable that small subsets will be accepted on the left side of a proposed split, as per the discussion in Section 2.2.

### 3.2 Sampling

For each experiment (exp-$c$ and exp-$v$), we drew samples for each pair $\mathcal{S}_{i,j}$ by sampling the appropriate BNs. For instance, for $\mathcal{S}_{2,2}$ and exp-$c$ we first drew a number of samples from $R_1$ in Figure 4, then a number of samples from $R_2$, and finally another number of samples from $R_1$. The number of samples drawn for each regime was picked at random from a uniform distribution between 50 and 70.

### 3.3 Methodology

Given a sample, we processed each data point from the sample in turn and ran the proposed algorithm. Each time a new data point $d_i$ was made available for the algorithm, we calculated the log-likelihood of this data point given the current regime BN in $M_C$, and then incorporated $d_i$ into $\mathcal{D}$ and ran the rest of the algorithm. In parallel, we also calculated the log-likelihood of $d_i$ given a single BN learnt using all previous data, as well as the log-likelihood of $d_i$ using the proposed algorithm, but without the `MERGE` procedure. Thus for each sample we had three sets of log-likelihoods: $L_s$, $L_{nm}$, $L_m$ (single, no merge and merge), along with the mean values of these sets: $\bar{L}_s$, $\bar{L}_{nm}$, $\bar{L}_m$. The following analysis was done for each experiment and pair $\mathcal{S}_{i,j}$:

- 50 samples were drawn from $\mathcal{S}_{i,j}$, resulting in three sets of means: $\{\bar{L}_s\}_1^{50}$, $\{\bar{L}_{nm}\}_1^{50}$ and $\{\bar{L}_m\}_1^{50}$.
- Two null hypotheses were stated:
  - $\mathcal{H}_0^1$: The differences between pairs of means in $\{\bar{L}_s\}_1^{50}$ and $\{\bar{L}_{nm}\}_1^{50}$ follow a symmetric distribution centred at zero.
  - $\mathcal{H}_0^2$: The differences between pairs of means in $\{\bar{L}_{nm}\}_1^{50}$ and $\{\bar{L}_m\}_1^{50}$ follow a symmetric distribution centred at zero.
- The Wilcoxon signed-rank test was used to test each null hypothesis. Two-tailed $p$-values below 0.01 were required to reject the null hypothesis.

### 3.4 Results and Discussion

In Table 2 we present the results from the exp-$c$ experiment. The first column gives the pair $\mathcal{S}_{i,j}$ under consideration, and the following three columns represent the median of the three sets $\{\bar{L}_s\}_1^{50}$, $\{\bar{L}_{nm}\}_1^{50}$ and $\{\bar{L}_m\}_1^{50}$. The final two columns gives the $p$-values for the two Wilcoxon signed-rank tests. Table 3 gives the same values for the exp-$v$ experiment.

In both experiments, when there were no regime shifts ($\mathcal{S}_{0,1}$), learning a single BN was significantly better than using the proposed learning algorithm. It is possible that a series of closely located outliers may result in the algorithm deciding to introduce a split in the dataset. However, the single BN never introduces this split, and it therefore has more data to estimate the parameters of the joint distribution of the single regime.

| Pair | Median $\{\bar{L}_s\}_1^{50}$ | Median $\{\bar{L}_{nm}\}_1^{50}$ | Median $\{\bar{L}_m\}_1^{50}$ | $p$-value $\mathcal{H}_0^1$ | $p$-value $\mathcal{H}_0^2$ |
|---|---|---|---|---|---|
| $\mathcal{S}_{0,1}$ | **-3.350** | -3.487 | -3.487 | $< 0.001^*$ | 1.0 |
| $\mathcal{S}_{1,1}$ | -3.792 | **-3.220** | **-3.220** | $< 0.001^*$ | 1.0 |
| $\mathcal{S}_{2,1}$ | -4.063 | **-3.155** | -3.160 | $< 0.001^*$ | 0.955 |
| $\mathcal{S}_{2,2}$ | -3.699 | -3.390 | **-3.323** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,1}$ | -5.484 | **-3.296** | -3.299 | $< 0.001^*$ | 0.7865 |
| $\mathcal{S}_{3,2}$ | -3.987 | -3.482 | **-3.298** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,3}$ | -4.020 | -3.212 | **-3.060** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,4}$ | -4.095 | -3.477 | **-3.196** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,5}$ | -3.717 | -3.214 | **-3.083** | $< 0.001^*$ | $< 0.001^{**}$ |

$*$ rejection of $\mathcal{H}_0^1$, $**$ rejection of $\mathcal{H}_0^2$

Table 2: Results from the exp-$c$ experiment

| Pair | Median $\{\bar{L}_s\}_1^{50}$ | Median $\{\bar{L}_{nm}\}_1^{50}$ | Median $\{\bar{L}_m\}_1^{50}$ | $p$-value $\mathcal{H}_0^1$ | $p$-value $\mathcal{H}_0^2$ |
|---|---|---|---|---|---|
| $\mathcal{S}_{0,1}$ | **-3.350** | -3.487 | -3.487 | $< 0.001^*$ | 1.0 |
| $\mathcal{S}_{1,1}$ | -3.840 | **-3.516** | -3.524 | $< 0.001^*$ | 0.1003 |
| $\mathcal{S}_{2,1}$ | -5.100 | -3.939 | **-3.933** | $< 0.001^*$ | 0.0267 |
| $\mathcal{S}_{2,2}$ | -3.935 | -3.561 | **-3.465** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,1}$ | -5.399 | **-3.861** | -3.967 | $< 0.001^*$ | 0.1231 |
| $\mathcal{S}_{3,2}$ | -4.790 | -3.691 | **-3.609** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,3}$ | -5.084 | -3.943 | **-3.895** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,4}$ | -5.051 | -4.129 | **-3.939** | $< 0.001^*$ | $< 0.001^{**}$ |
| $\mathcal{S}_{3,5}$ | -3.838 | -3.407 | **-3.155** | $< 0.001^*$ | $< 0.001^{**}$ |

$*$ rejection of $\mathcal{H}_0^1$, $**$ rejection of $\mathcal{H}_0^2$

Table 3: Results from the exp-$v$ experiment

However, for all systems that exhibit regime changes, the proposed algorithm was significantly better than using a single BN. This was even true in the exp-$c$ experiment, where the BNs of the different regimes retained more similarity. This result is very encouraging, as in a real world setting we would not be able to test how much the BNs change between regimes, however as we have shown here, the proposed algorithm is sensitive to large as well as small changes.

The transition structures for $\mathcal{S}_{1,1}$, $\mathcal{S}_{2,1}$ and $\mathcal{S}_{3,1}$ are all chains of regimes. As expected, in both the exp-$c$ and exp-$v$ experiments there was no significant difference between using the proposed algorithm with or without the MERGE procedure for these pairs. However, for all other pairs the difference was significant, with a lower median when including the MERGE procedure. When regimes reoccur, we can get better estimates of the BNs' parameters by using all data belonging to a specific regime, as is evident from the increased performance when using the MERGE procedure.

All in all, these baseline results confirms that the algorithm works as intended, adapting to regime shifts, and thereby suppling a BN which better represents the current regime than a BN learnt using all the available data.

## 4. Real World Application

The baseline results from the experiments in Section 3 are very promising, as they confirm that the proposed algorithm performs as expected. This prompted us to investigate the performance of the algorithm on real world data. To this end we created two datasets, $US$ and $EU$, with data from the financial domain. Each dataset contained five variables which represented the *daily price volatility*

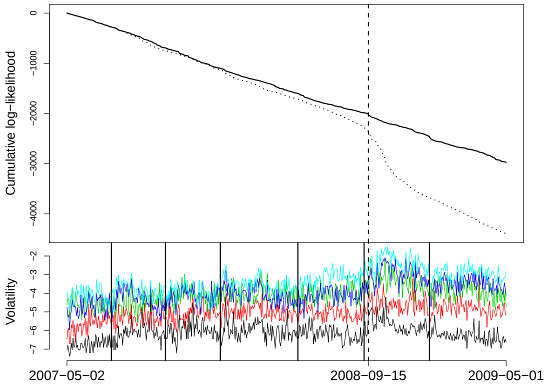


Figure 6: Cumulative log-likelihood and splits for the $US$ dataset

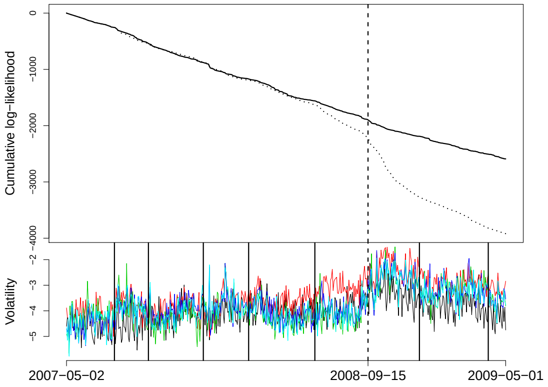


Figure 7: Cumulative log-likelihood and splits for the $EU$ dataset

of financial assets. We will omit the full specification and calculation of the price volatility, and instead refer the interested reader to Bendtsen and Peña (2016c). It will suffice to understand the price volatility as a measure for the span over which the price of an asset ranges on a specific day, thus a large span means an increased price volatility. The variables in the two datasets represented the daily price volatility of the following assets:

- $US$: short-term US debt, long-term US debt, stocks of US companies, gold and stocks of companies related to oil and gas.
- $EU$: stocks of French companies, stocks of German companies, stocks of UK companies, gold and stocks of companies related to oil and gas.

For each dataset we ran the proposed learning algorithm and calculated the log-likelihood for each new datapoint (similar to the procedure described in Section 3.3). We then calculated the cumulative sum of these log-likelihoods. The series are plotted in Figure 6 for $US$ and Figure 7 for $EU$. In the figures, the solid line represents the proposed algorithm and the dotted line a single BN. Below the cumulative log-likelihoods we have plotted the data for each variable, along with the splits introduced by the algorithm. In this setting, data is made available on a daily basis, thus one iteration of the algorithm needs to be run each day, given the previous day's model. In our experiments, the average time to complete one iteration was approximately two minutes, thus giving the algorithm ample time to complete during the day. However, if working in a domain where data is more frequent than the time that the algorithm needs to complete, then a mini-batch mode framework or similar solution needs to be considered.

It is clear that up until September 2008, the performance of the two approaches are approximately equal, however past this date the single BN's performance deteriorates compared to the proposed algorithm. During the spring of 2007 it was becoming evident that some of the mortgage backed securities that were being traded, and used as collateral, were not as safe as previously thought. Bear Stearns liquidated two of its hedge funds in late July, and BNP Paribas froze three of their funds in August, as they were unable to price them. These issues escalated in 2008, and in September two major events occurred in the US financial system: first on 2008-09-07 the US government decided to bail out Fannie Mae and Freddie Mac, two firms that were guaranteeing a large part of the notorious *sub-prime* mortgage market, and then Lehman Brothers filed for bankruptcy on 2008-09-15. It was after these two events that the growing US financial crisis became a full blown global financial crisis. Figure 6 and Figure 7 suggest that there were regime shifts among the price volatilities measured during this period, and that the proposed algorithm is able to adapt to these shifts, since the cumulative log-likelihood does not dramatically change but rather seems to decrease at the same rate.

## 5. Conclusions and Summary

We have proposed a regime aware learning algorithm for learning a sequence of BNs in order to adapt to regime changes in the underlying system being modelled. In order to evaluate the feasibility of the algorithm, we created a baseline test against learning a single BN. Our experiments show that the proposed algorithm significantly outperforms the single BN, even in cases where there are similarities between the BNs of the different regimes. We have also shown that by exploiting the fact that regimes may reoccur, the algorithm's performance can be further improved. Our experiments on real world data suggest that the algorithm is able to adapt to regime changes in financial data, allowing the investigator to have a BN at their disposal that better reflects the current regime of the modelled system.

In this paper we have only concerned ourselves with determining that the proposed algorithm works as expected. From here there are several important aspects that will require further investigation. For instance, a comparison of our proposed algorithm with other approaches, e.g. that of Nielsen and Nielsen (2008), would inform us of cases where the approaches may outperform each other. Furthermore, while in this paper we have focused on the BN of the current regime, the entire sequence of BNs that is generated can be informative, and may be combined into a gated Bayesian network (Bendtsen and Peña, 2016a) in such a way that it may be possible to predict which regime the underlying system may transition into next, and how long it will take for this transition to happen. We are also interested in attempting to remove any uncertainty about the effect of the BN structure learning algorithm, potentially using exact structure learning (Yuan and Malone, 2013; Sonntag et al., 2015).

## References

T. Andersen, J. Carstensen, E. Hernandez-Garcia, and C. M. Duarte. Ecological thresholds and regime shifts: approaches to identification. *Trends in Ecology & Evolution*, 24(1):49–57, 2009.

S. Bach and M. Maloof. A Bayesian approach to concept drift. In *Advances in Neural Information Processing Systems 23*, pages 127–135, 2010.

M. Bendtsen and J. M. Peña. Gated Bayesian networks for algorithmic trading. *International Journal of Approximate Reasoning*, 69:58–80, 2016a.

M. Bendtsen and J. M. Peña. Regimes in baseball player's career data. *Data Mining and Knowledge Discovery*, 2016b. Under review.

M. Bendtsen and J. M. Peña. Detecting regime changes with gated Bayesian networks. *Journal of Artificial Intelligence Research*, 2016c. Under review.

H. Borchani, A. M. Martínez, A. R. Masegosa, H. Langseth, T. D. Nielsen, A. Salmerón, A. Fernández, A. L. Madsen, and R. Sáez. Modeling concept drift: a probabilistic graphical model based approach. In *Advances in Intelligent Data Analysis XIV: 14th International Symposium*, pages 72–83, 2015.

W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60, 1991.

N. Friedman and M. Goldszmidt. Sequential update of Bayesian network structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 165–174, 1997.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, March 1995.

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

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

W. Lam. Bayesian network refinement via machine learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):240–251, 1998.

W. Lam and F. Bacchus. Using new data to refine a Bayesian network. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 383–390, 1994.

R. M. May, S. A. Levin, and G. Sugihara. Complex systems: ecology for bankers. *Nature*, 451 (7181):893–895, 2008.

S. H. Nielsen and T. D. Nielsen. Adapting Bayes network structures to non-stationary domains. *International Journal of Approximate Reasoning*, 49(2):379–397, 2008.

M. Pal, A. K. Pal, S. Ghosh, and I. Bose. Early signatures of regime shifts in gene expression dynamics. *Physical Biology*, 10(3):036010, 2013.

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

M. Scheffer, S. Carpenter, J. A. Foley, C. Folke, and B. Walkerk. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596, 2001.

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

C. Yuan and B. Malone. Learning optimal Bayesian networks: a shortest path perspective. *Journal of Artificial Intelligence Research*, 48(1):23–65, Oct 2013.