Autoregressive Asymmetric Linear Gaussian Hidden Markov Models

Carlos Puerto-Santana, Pedro Larrañaga and Concha Bielza

Abstract—In a real life process evolving over time, the relationship between its relevant variables may change. Therefore, it is advantageous to have different inference models for each state of the process. Asymmetric hidden Markov models fulfil this dynamical requirement and provide a framework where the trend of the process can be expressed as a latent variable. In this paper, we modify these recent asymmetric hidden Markov models to have an asymmetric autoregressive component in the case of continuous variables, allowing the model to choose the order of autoregression that maximizes its penalized likelihood for a given training set. Additionally, we show how inference, hidden states decoding and parameter learning must be adapted to fit the proposed model. Finally, we run experiments with synthetic and real data to show the capabilities of this new model.

Index Terms—Hidden Markov models, Bayesian networks, Model selection, Structure learning, Time series, Information asymmetries, Linear Gaussian, Autoregressive, Yule-Walker equations.

✦

1 INTRODUCTION

HIDDEN Markov models (HMMs) have been successfully used to analyze dynamic signals, e.g., in speech recognition [1] and tool wearing monitoring [2] or sequential signals, e.g., in gene prediction [3]. These models assume the existence of a latent or hidden variable that drives an observable set of variables. However, traditional HMMs in the case of continuous data, make the hypothesis that for all the driving dynamical process, a complete dependence probabilistic model involving all the variables is held, which can be untrue. This causes that the models learn a considerable number of unnecessary parameters that may cause data overfitting.

The idea of asymmetric HMMMs is introduced in [4] and [5]. These models imply that depending on the value of certain variables, the distribution of the remaining variables may change. For HMMs, the asymmetric component is expressed with the hidden variable, with which depending on its value, a context-specific Bayesian network [6] encodes the distribution of the emission probabilities. These context-specific Bayesian networks reduce the number of parameters needed.

Autoregressive (AR) processes have been studied for a long time, especially for regression tasks [7]. However, the traditional approaches to AR processes make strong assumptions as to stationariness that do not hold for many real case scenarios. This issue was addressed by [8], allowing the models to have changing parameters depending on the value of a hidden variable. Nevertheless the order of the AR process had to be fixed beforehand by trial and error.

HMMs and AR processes were combined in [9], where AR coefficients were added to the emission probabilities.

In this paper we combine the ideas of asymmetric HMMs with AR processes to overcome the previous shortcomings: determine the AR order of a model for each hidden state and reduce the number of unnecessary parameters. Specifically, our model enables each variable, depending on the hidden state, to determine its parents within the context-specific Bayesian network and the number of lags that its distribution requires to maximize a model fitting score.

The structure of this document is as follows. Section 2 describes related work about asymmetric probabilistic models and HMMs with AR processes. Section 3 reviews HMMs in general and summarizes the expectation maximization algorithm (EM), the structural EM and the Yule-Walker equations [7] that are relevant tools for our model. Section 4 introduces the proposed autoregressive asymmetric linear Gaussian hidden Markov model (AR-AsLG-HMM). In this section we discuss the adaptation of the forward-backward and Viterbi algorithms [1]. We also describe the parameter and structural learning and show that the EM algorithm iteratively improves the log-likelihood of the data for our model. Section 5 presents experiments with synthetic data, real air quality and ball-bearing degradation data. The results obtained using the AR-AsLG-HMM are compared against its non-AR version and other state-of-the-art approaches. The paper is rounded off in Section 6 with conclusions and comments regarding possible future research.

- C. Puerto-Santana, Universidad Politécnica de Madrid, Spain
  E-mail: ce.puerto@alumnos.upm.es
- C. Puerto-Santana, Aingura IIoT, Spain
  E-mail: epuerto@ainguraiiot.com
- P. Larrañaga, Universidad Politécnica de Madrid, Spain
  E-mail: pedro.larranaga@fi.upm.es
- C. Bielza , Universidad Politécnica de Madrid, Spain
  E-mail: mcbielza@fi.upm.es

2 RELATED WORK

In this section we review the related work regarding HMMs with AR behavior and asymmetric probabilistic models. Table 1 shows the reviewed articles grouped according to their contribution.

Modified emission probabilities in HMMs:
AR polynomials in emission probabilities [9]
AR Mixture of Gaussians HMM (AR-MoG-HMM) [10]
Markov mean-switching AR model (MMSAR) [11]
Vector AR multivariate Gaussian HMM (VAR-MVGHMM) [12]
Linear Markov switching AR model (LMSAR) [8]
Gaussian AR-HMMs with a linear error coefficient [13]
AR hidden semi-Markov model (AR-HSMM) [14]
Transitional Markov switching autoregressive model (TMSAR) [15]
Vector AR hierarchical HSMM (VAR-HHSMM) [16]

Modified hidden variables:
AR-HMM with an additional memoryless hidden variable [17]
Higher-order AR-HMM (AR-HO-HMM) [18]

Missing data in HMMs:
AR-HMM with a missing at random assumption [19]
AR-HMM with missing data as latent variables [20]

Asymmetric models:
Similarity networks [21]
Bayesian multinets [22]
Context-specific Bayesian networks [6]
Buried Markov model (BMM) [23]
Conditional Chow-Liu trees with HMMs [4]
Chain events graph (CEG) [24]
Stratified graphical model (SGM) [25]
Dynamic chain events graph [26]
Asymmetric HMM with discrete variables (As-HMM) [5]
Asymetric HMM with continuous variables (AsLG-HMM) [27]

TABLE 1
Reviewed articles and their contributions to asymmetric HMMs and AR HMMs

2.1 Modified emission probabilities in HMMs

One of the first combinations of HMM and AR models attempted to process speech data [9]. Autoregressive polynomials were added to the Gaussian emission probabilities, in which coefficients were determined via the Baum-Welch algorithm [1]. Later, [10] proposed mixtures of Gaussian hidden Markov models (AR-MoG-HMMs) where the emission probabilities were modelled as mixtures of Gaussians. These models were used for speech recognition. In [12] a vectorial AR multivariate Gaussian HMM (VAR-MVGHMM) was introduced. This model enables variables to have temporal dependencies with all the other variables. Again, the model was used for speech recognition.

Some authors [13] modified the emission probabilities such that they behave as an AR Gaussian but with an error coefficient given by the linear prediction residuals [28].

Others also considered variations of HMMs such as hidden semi-Markov models (HSMMs), where the time duration of each hidden state can be modified to not always follow a geometric distribution, or hierarchical hidden Markov models (HHMMs) where AR behavior was considered. For instance, [14] proposed an AR-HSMM, where AR variables and non-AR variables could be considered in the same model depending on the modeller's decision. [16] proposed a vector AR hierarchical hidden semi-Markov model (VAR-HHSMM) to classify and determine hand movements.

Other approximations of HMMs with AR properties can be found in [11] and [8]. The author proposed an edited log-likelihood function to represent the AR behavior in data. Markov mean-switching AR models (MMSAR) and linear Markov-switching AR model (LMSAR) were studied and their parameters were calculated with the EM algorithm. [15] proposed the transitional Markov switching autoregressive (TMSAR) model as an extension of MMSAR and LMSAR models. In this case, the emission probabilities depend

on past values of the hidden process to determine changes in its mean and its weight. The authors used maximum likelihood methods with a Newton-Raphson strategy to estimate the model parameters.

2.2 Modified hidden variables

In more recent works, new approaches have been proposed in which the assumptions about the hidden variables that govern the process were modified such as the model given by [17], where the authors edited an autoregressive hidden Markov model (AR-HMM) by introducing a memoryless hidden variable. The Markovian hidden states had a probabilistic dependency of this memoryless hidden variable. AR higher-order HMMs (AR-HO-HMMs) were introduced in [18]. The authors not only considered an autoregressive property in the observations, but also a fixed order Markov assumption in the hidden states specified by the user. They used mixtures of Gaussians with AR properties for the emission probabilities.

2.3 Missing data in HMMs

Other works focused on the missing data. In [19] an AR-HMM with a missing at random assumption was proposed to perform exact inference in such scenarios. In [20] the missing data was considered as latent variables. Specially, when the sampling rate of the signal was not high enough, hidden variables were added between observations. Additionally, the authors proposed a modified forward-backward algorithm and Baum-Welch parameter updating formulas.

2.4 Asymmetric models

Regarded as asymmetric probabilistic graphical models, the Bayesian multinets introduced in [22] were used to describe different local graphical models depending on the values of certain observed variables; the similarity networks in [21] allowed the creation of independent influence diagrams[1] for subsets of a given domain. Context-specific independence in Bayesian networks in [6] used tree structured conditional probability distributions with a D-separation-based algorithm to determine statistical dependencies between variables according to contexts given by instantiations of subsets of variables. Following these ideas, more recently in [25], stratified graphical models (SGM) were proposed, where the concept of stratum was introduced to allow different factorizations for a probability distribution depending on the values of some of the variables. A nonreversible Metropolis-Hastings algorithm to calculate marginal likelihoods and learn decomposable SGMs was given. [24] introduced the chain events graphs (CEG). A CEG consists of a directed colored graph obtained from a staged tree[2] by successive edge contraction operations. The obtained graphical model can represent conditional independence and causal behavior

1. An influence diagram is a probabilistic graphical model used for decision problems, where random, decision and value nodes are present [29]

2. A staged tree is a probabilistic graphical model, where the graph is a tree and the nodes are random variables whose non leaf variables are identified with the same color if they have the same conditional probabilistic relationships with their children nodes [24]

that traditional Bayesian networks cannot show. Later, a dynamic version was proposed [26].

Other authors have attempted to combine asymmetric models with HMMs. For example, in [23] the buried Markov models (BMM) were introduced. In these articles, the models of [12] were used, but the temporary dependencies can vary depending on the hidden state. These context-specific dependencies are learned using mutual information strategies. [4] used Chow-Liu trees and conditional Chow-Liu trees coupled with HMMs. The HMMs were used to model the dynamic behavior of a process, and the Chow-Liu tree was used to model the emission probabilities. A Chow-Liu tree or conditional Chow-Liu tree was associated with each value of the hidden variable. The parameters of the model were computed with the EM algorithm; specifically, the tree structure was determined in the maximization step. However, the model was specified only for discrete variables. More recently, asymmetric hidden Markov models (As-HMMs) were proposed in [5], where a local graphical model was associated with each value of the hidden variable, and the graphical model was not restricted to Chow-Liu trees. However, again only models with discrete observable variables were allowed. In [27], this issue was addressed with the asymmetric linear Gaussian HMMs (AsLG-HMMs), where the emission probabilities were modeled as conditional linear Gaussian Bayesian networks. The estimation of the model parameters was performed with the EM algorithm.

In this paper, we extend asymmetric HMMs for continuous variables of [27], where the model during its learning phase can estimate for each variable the order of the AR process as well as its parameters depending on the context or value of the hidden variable. Thus, we couple for the first time asymmetric linear Gaussian HMMs with AR processes.

3 THEORETICAL FRAMEWORK

Because the proposed model needs to fit the forward-backward and Viterbi algorithms, we first review these algorithms and the traditional HMM. The parameter and structure learning of the proposed model will be performed via the EM and SEM algorithms; therefore, we also review these algorithms and their properties. Additionally, because the Yule-Walker equations will be used to determine the order of an AR process, they are briefly examined. Additionally, in Table 2, a description of relevant symbols used in this article is shown.

3.1 Hidden Markov Models

An HMM can be seen as a double chain stochastic model, where a chain is observed, namely $\boldsymbol{X}^{0:T} = (\boldsymbol{X}^0, ..., \boldsymbol{X}^T)$, where $\boldsymbol{X}^t = (X_1^t, ..., X_M^t) \in \mathcal{R}^M$ and the other chain is hidden, namely $\boldsymbol{Q}^{0:T} = (Q^0, ..., Q^T)$. Here, T is the length of the data. The usual approach for HMMs [1] is to assume that the hidden process has the first-order Markovian property, that is, $P(Q^t|\boldsymbol{Q}^{0:t-1}) = P(Q^t|Q^{t-1})$. Furthermore, it is assumed that the observable process depends on the hidden process, more specifically $P(\boldsymbol{X}^t|\boldsymbol{X}^{0:t-1}, \boldsymbol{Q}^{0:t}) = P(\boldsymbol{X}^t|Q^t)$. Additionally it is assumed that the range R of the hidden variable is finite, i.e., $R(Q^t) = \{1, 2, ..., N\}$ for $t = 0, 1, ..., T$. Moreover $\boldsymbol{R}(\boldsymbol{Q}^{0:T}) = \{1, 2, ..., N\}^{T+1}$.

Symbol	Meaning
N	Number of hidden states
M	Number of variables
$\boldsymbol{Q}^{0:T}$	Sequence of hidden states from time 0 up time T
$\boldsymbol{X}^{0:T}$	Sequence of observations from time 0 up time T
$R(\cdot)$	Range of a random variable
$\mathbf{A}$	Transition matrix
a_{ij}	Transition probability of hidden state i to j
$\boldsymbol{\pi}$	Initial probability distribution
π_i	Probability of starting at hidden state i
$\mathbf{B}$	Emission probabilities
$b_i^{p^*}(\boldsymbol{x}^t)$	Emission probability for the proposed model
Ω	Space of model parameters
σ_{im}^2	Variance of the Gaussian of variable X_m at state i
$\boldsymbol{\lambda}$	Model parameters
$\boldsymbol{\lambda}'$	Prior parameters
$\alpha^t(i)$	Forward variable at time t for the hidden state i
$\alpha_{p^*}^t(i)$	Forward variable for the proposed model
$\beta^t(i)$	Backward variable at time t for hidden state i
$\beta_{p^*}^t(i)$	Backward variable for the proposed model
$\delta^t(i)$	Most probable sequence of hidden states up to time $t-1$
$\delta_{p^*}^t(i)$	Most probable sequence of hidden states for the proposed model
$\psi^t(i)$	Most probable transition from hidden state i at time t
$\gamma^t(i)$	Probability of hidden state i at time t
$\xi^t(i,j)$	Transition probability from hidden state i to j at time t
$\mathcal{Q}$	Auxiliary optimization function
$\mathcal{Q}^{p^*}$	Auxiliary optimization function for the proposed model
$\mathcal{B}$	Probabilistic graphical model
$\mathcal{B}'$	Prior probabilistic graphical model
$\#(\cdot)$	Number of parameters of the input graphical model
ϕ_{kj}	Weight of the j lag when k lags are used
ρ_k	Correlation between X^t and X^{t-k}
Φ_k	ρ_k but removing intermediary lags effect
$E[\cdot]$	Expectation operator
p^*	Maximum admissible lag
$\mathbf{Pa}_i(X_m)$	The set of parents of X_m for hidden state i
$\mathbf{pa}_{im}^t$	Value at t of $\mathbf{Pa}_i(X_m)$ for hidden state i
U_{imk}	k^{th} parent of variable X_m at hidden state i
k_{im}	Number of fathers for variable X_m at the hidden state i
$\boldsymbol{\beta}_{im}$	Weights of $\mathbf{pa}_{im}^t$ for the linear dependency of X_m
$\mathbf{d}_{im}^t$	Value at t of the AR variables of X_m for hidden state i
$\boldsymbol{\eta}_{im}$	Weights of $\mathbf{d}_{im}^t$ for the linear dependency of X_m
f_{im}^t	$\boldsymbol{\beta}_{im} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im} \cdot \mathbf{d}_{im}^t$
$g(i)$	Numeric label to hidden state i
$\boldsymbol{\kappa}$	Standard values for labelling
$\mathbf{v}$	Scaling vector for labelling

TABLE 2
Symbols used in Section 3 and Section 4

All the previous HMM specifications can be summarized with the parameter $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) \in \Omega$, where Ω denotes the space of all possible parameters, $\mathbf{A} = [a_{ij}]_{i,j=1}^N$ is a matrix representing the transition probabilities between hidden states $i, j \in R(Q^t)$ over time, i.e., $a_{ij} = P(Q^{t+1} = j|Q^t = i, \boldsymbol{\lambda})$; $\mathbf{B}$ is a vector representing the emission probability of the observations given the hidden state, $\mathbf{B} = [b_i(\mathbf{x}^t)]_{i=1}^N$, where $b_i(\boldsymbol{x}^t) = P(\boldsymbol{X}^t = \boldsymbol{x}^t|Q^t = i, \boldsymbol{\lambda})$ is a probability density function; $\boldsymbol{\pi}$ is the initial probability distribution of the hidden states, $\boldsymbol{\pi} = [\pi_j]_{j=1}^N$, where $\pi_j = P(Q^0 = j|\boldsymbol{\lambda})$.

Additionally, an HMM can be seen as a probabilistic graphical model [30] (Fig. 1), where the nodes of the graph represent random variables and the arcs represent direct probabilistic dependencies.

Three main tasks can be performed in the context of HMMs. First, compute the likelihood of an observation $\boldsymbol{x}^{0:T}$ given a model $\boldsymbol{\lambda}$, i.e., $P(\mathbf{x}^{0:T}|\boldsymbol{\lambda})$, which can be performed using the forward-backward algorithm. Second, compute the most likely sequence of hidden states and observations,
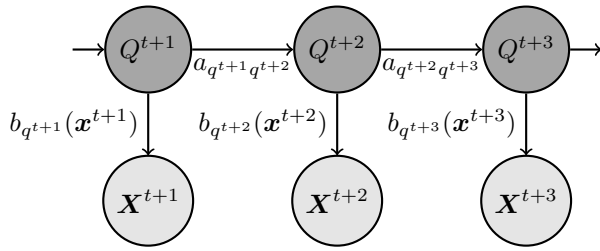
Fig. 1. An HMM as a probabilistic graphical model

i.e., find the value of $\delta^t(i) = \max_{\boldsymbol{q}^{0:t-1}}\{P(\boldsymbol{x}^{0:t}, \boldsymbol{q}^{0:t-1}, Q^t = i|\boldsymbol{\lambda})\}$, $t = 0, ..., T$, $i = 1, ..., N$, which can be solved using the Viterbi algorithm. Third, learn the parameter $\boldsymbol{\lambda}$, which is estimated with the EM algorithm. A theoretical tutorial for understanding these algorithms can be found in [1]. We briefly review them below.

3.2 The forward-backward algorithm

To execute the forward-backward algorithm, we first must define the forward and backward variables: $\alpha^t(i) = P(Q^t = i, \boldsymbol{x}^{0:t}|\boldsymbol{\lambda})$, $\beta^t(i) = P(\boldsymbol{x}^{t+1:T}|Q^t = i, \boldsymbol{\lambda})$, respectively, $i = 1, ..., N$, $t = 0, ..., T$. The forward and backward variables can be written recursively:

$$\alpha^{t+1}(i) = \sum_{j=1}^{N} b_i(\boldsymbol{x}^{t+1})a_{ji}\alpha^t(j)$$

$$\beta^t(i) = \sum_{j=1}^{N} b_j(\boldsymbol{x}^{t+1})a_{ij}\beta^{t+1}(j)$$

Their initial values are $\alpha^0(i) = \pi_i b_i(\boldsymbol{x}^0)$ and $\beta^T(i) = 1$. The forward variable can help us compute the likelihood of $\boldsymbol{x}^{0:T}$ since:

$$P(\boldsymbol{x}^{0:T}|\boldsymbol{\lambda}) = \sum_{i=1}^{N} P(\boldsymbol{x}^{0:T}, Q^T = i|\boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha^T(i).$$

3.3 The Viterbi algorithm

Variable $\delta^t(i)$ for time $t = 0, ..., T$ and hidden state $i = 1, ..., N$ can be written as:

$$\delta^t(i) = \max_{j=1,...,N}\{a_{ji}\delta^{t-1}(j)\}b_i(\boldsymbol{x}^t).$$

Its initial value is $\delta^0(i) = \pi_i b_i(\boldsymbol{x}^0)$. However, to find the most likely sequence of states $\boldsymbol{q}^{0:T}$, it is necessary to iteratively calculate an auxiliary variable $\psi^t(i) = \arg\max_{j=1,...,N}\{\delta^{t-1}(j)a_{ji}\}$, $i = 1, ..., N$, $t = 0, ..., T$, which records the most likely transitions between states. Then, a backtracking process must be performed to recover $\boldsymbol{q}^{0:T}$, taking $q^T = \arg\max_{i=1,...,N}\{\delta^T(i)\}$ and $q^t = \psi^{t+1}(q^{t+1})$ for $t = T - 1, ..., 0$.

3.4 The EM algorithm

To learn the parameter $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ given a dataset $\boldsymbol{x}^{0:T}$ and a priori $\boldsymbol{\lambda}'$, the traditional EM approach [31] is used. In the EM algorithm, two steps called the expectation step (E-step) and maximization step (M-step) are iterated until convergence is met.

For the E step, we will need only to calculate the probabilities $\gamma^t(i) := P(Q^t = i|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}')$ and $\xi^t(i,j) := P(Q^t = i, Q^{t+1} = j|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}')$ $i, j = 1, ..., N$, $t = 0, ..., T$, which are related in the following manner: $\sum_{j=1}^{N} \xi^t(i,j) = \gamma^t(i)$.

For the M step, we must derive the updating formulas for parameter $\boldsymbol{\lambda}$. For π_i and a_{ij} for the hidden states $i, j = 1, ..., N$, are:

$$\pi_i^* = \gamma^0(i),$$
$$a_{ij}^* = \frac{\sum_{t=0}^{T-1} \xi^t(i,j)}{\sum_{t=0}^{T} \gamma^t(i)}.$$

The updating formula for parameter $\mathbf{B}$ relies on the assumptions made over the transition and emission probabilities. For example, in [1], the updating formulas are calculated when the emission probabilities are assumed to be discrete, a mixture of Gaussians (MoG) or a mixture of AR Gaussians (AR-MoG). If the hypotheses about the transition probabilities or the initial distribution change, the formulas given above are no longer valid.

3.5 The SEM algorithm

When we deal with an unknown a priori probabilistic graphical model $\mathcal{B}$, it is desirable to find the structure that maximizes the likelihood of the data. However, as many parameters are used in dense networks, the likelihood improves but it can be due to data overfitting. Therefore, penalized likelihood-based scores such as the Bayesian information criterion (BIC) or Akaike information criterion are used in structure optimization algorithms to prevent this issue. In [32], the structural EM (SEM) algorithm is introduced with its convergence and optimality properties. SEM finds both the desired model and the parameters. SEM tries to maximize the function $\mathcal{Q}(\mathcal{B}, \boldsymbol{\lambda}|\mathcal{B}', \boldsymbol{\lambda}')$, where $\mathcal{B}'$ is a previous or a prior graphical model:

$$\mathcal{Q}(\mathcal{B}, \boldsymbol{\lambda}|\mathcal{B}', \boldsymbol{\lambda}') = E_{P(\boldsymbol{q}^{0:T}|\boldsymbol{x}^{0:T}, \mathcal{B}', \boldsymbol{\lambda}')}[\ln P(\boldsymbol{x}^{0:T}, \boldsymbol{q}^{0:T}|\mathcal{B}, \boldsymbol{\lambda})] - 0.5\#(\mathcal{B})\ln(T). \tag{1}$$

Eq. (1) considers changes in the structure of the probabilistic graphical model and its parameters, and to prevent overfitting, it is penalized by the number of parameters in the model $\#(\mathcal{B})$ and the logarithm of the length of the data T. The SEM algorithm consists of using the EM algorithm with a prior model, $\mathcal{B}'$, and a prior parameter $\boldsymbol{\lambda}'$ to obtain the parameters $\boldsymbol{\lambda}''$; then, using the latent probabilities $P(\boldsymbol{q}^{0:T}|\boldsymbol{x}^{0:T}, \mathcal{B}', \boldsymbol{\lambda}'')$ and the parameters $\boldsymbol{\lambda}''$ finding a new structure $\mathcal{B}''$ by solving $\max_{\mathcal{B}} \mathcal{Q}(\mathcal{B}, \boldsymbol{\lambda}''|\mathcal{B}', \boldsymbol{\lambda}'')$. Finally, the EM is again applied to the new structure. This process is iterated until convergence is met.

3.6 The Yule-Walker equations

The Yule-Walker equations [7] will be a key issue in constructing the proposed model. A linear AR process with k time lag coefficients for a one-dimensional variable Y^t can be described as:

$$Y^t = \phi_{k1}Y^{t-1} + \cdots + \phi_{kk}Y^{t-k} + \epsilon^t \tag{2}$$

where $\epsilon^t \sim \mathcal{N}(0, \sigma^2)$ is an error term following a Gaussian distribution with mean zero and variance σ^2. Correlogram

function ρ_k returns the correlation between Y^t and Y^{t-k}. We define $\bar{Y}^t := Y^t - \mu_Y$ where μ_Y is the mean of Y^t and $\zeta_k := E[\bar{Y}^t \bar{Y}^{t-k}]$, which is the expected value of the product of both shifted variables. The correlogram function is computed as:

$$\rho_k. := \frac{\zeta_k}{\zeta_0}$$

The partial correlogram function $\Phi(k)$ encodes the correlation between variables Y^t and Y^{t-k} once the effect from intermediary lags has been removed. To determine these partial correlations, observe that for $l \in \{1, ..., k\}$:

$$\bar{Y}^t = \phi_{k1}\bar{Y}^{t-1} + \cdots + \phi_{kk}\bar{Y}^{t-k} + \epsilon^t$$
$$\bar{Y}^t\bar{Y}^{t-l} = \phi_{k1}\bar{Y}^{t-1}\bar{Y}^{t-l} + \cdots + \phi_{kk}\bar{Y}^{t-k}\bar{Y}^{t-l} + \epsilon^t\bar{Y}^{t-l}$$
$$\rho_l = \phi_{k1}\rho_{l-1} + \cdots + \phi_{kk}\rho_{k-l}$$
$$(3)$$

In the last line of Eq. (3), we applied the expectation operator and divided by ζ_0. We assumed that $E[\bar{Y}^{t-l}\epsilon^t] = 0$ for all t, which implies that Y^t is not correlated with the error term; a plausible hypothesis in real situations. Additionally, $\rho(0) = 1$. Moreover, notice that if these equations are computed for $l = 1, ..., k$, we obtain a system of linear equations, which corresponds to the Yule-Walker equations:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix} = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{k-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix}.$$

The partial correlogram function returns $\Phi(k) := \hat{\phi}_{kk}$. Note that if we wish to evaluate up to k lags for the partial correlogram function, we must construct and solve k linear systems.

Assume that the sample is white noise. Then the parameter $\hat{\phi}_{kk}$ is distributed approximately as $\mathcal{N}(0, 1/T)$. With this information, it is possible to perform hypothesis tests to determine the relevancy of each lag parameter. If $\Phi(p^*)$ is the higher time lag coefficient that is significantly different from zero, then p^* is considered the AR order of the model [7].

It is worth mentioning that the previous lag estimation is only useful when the observed data are stationary; in other words, the parameters do not change over time. This particular assumption is violated for the problems in which we want to use HMMs because identifying changes in the model parameters according to changes in the data distribution is sought. With our proposed model, however, we will see that the order of the AR process within the HMM will be able to change dynamically and self adapt depending on the state of the hidden variable.

4 PROPOSED MODEL

The proposed model uses context-specific linear Gaussian Bayesian networks to factorize the emission probabilities. The context is given by the hidden variable. Also, an AR component is added to each variable. The AR order of each variable for each possible context is determined by

the SEM algorithm and the Yule-Walker equations when a score (to be specified later on) is optimized. Furthermore, for the proposed model, the likelihood function is modified; therefore, the forward-backward, Viterbi and EM algorithms have to be adapted.

4.1 Autoregressive asymmetric linear Gaussian hidden Markov models

Let p_m^* be the AR order (time lag) determined by the Yule-Walker equations and the individual relevancy hypothesis tests for each variable $X_m^t, m = 1, ..., M$. Set $p^* = \max_m p_m^*$. For our proposed model we work with the following log-likelihood function which ensures that during the SEM algorithm, the updated structures and AR orders are comparable:

$$LL(\boldsymbol{\lambda}) = \ln P(\boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})$$
$$= \ln \sum_{\boldsymbol{q}^{0:T} \in \boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}, \boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda}). \quad (4)$$

For this proposed HMM model which is, as explained below, asymmetric autoregressive with linear Gaussian emission probabilities (AR-AsLG-HMM), we modify the emission probabilities $\{b_i(\boldsymbol{x}^t)\}_{i=1}^N$ such that they can be factorized into linear Gaussian Bayesian networks [33] with an asymmetric component [5], i.e., each variable X_m for each state $i \in R(Q)$ is associated with a set of parents $\mathbf{Pa}_i(X_m) = \{U_{im1}, ..., U_{imk_{im}}\} \subset \{X_1, ..., X_M\}$ of size k_{im} (apart from Q) which influences its mean in a linear form. Additionally, the emission probabilities are now conditional probabilities given $p_{im} \leq p^*$ past values of the variables $X_m^t, m = 1, ..., M$ (AR terms) for each state $i \in R(Q)$. More specifically, we define:

$$b_i^{p^*}(\boldsymbol{x}^t) = P(\boldsymbol{x}^t|Q^t = i, \boldsymbol{x}^{t-p^*:t-1}, \boldsymbol{\lambda})$$
$$= \prod_{m=1}^M P(x_m^t|Q^t = i, x_m^{t-p_{im}:t-1}, \mathbf{Pa}_i(X_m), \boldsymbol{\lambda})$$
$$= \prod_{m=1}^M \mathcal{N}(x_m^t|\boldsymbol{\beta}_{im} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im} \cdot \boldsymbol{d}_{im}^t, \sigma_{im}^2) \quad (5)$$

In Eq. (5), we have $\boldsymbol{\beta}_{im} = (\beta_{im0}, ..., \beta_{imk_{im}})$, $\mathbf{pa}_{im}^t = (1, u_{im1}^t, ..., u_{imk_{im}}^t)$, $\boldsymbol{\eta}_{im} = (\eta_{im1}, ..., \eta_{imp_{im}})$ and $\boldsymbol{d}_{im}^t = (x_m^{t-1}, ..., x_m^{t-p_{im}})$. Fig. 2 shows an example of an AR-AsLG-HMM. In this example, when $Q^t = 1$ (top), variable X_2^t is dependent on Q^t, X_1^t, X_2^{t-1} and X_2^{t-2}, but X_1^t depends only on Q^t and X_1^{t-1}. However, when $Q^t = 2$ (bottom), X_1^t depends only on Q^t and X_2^t is dependent on X_2^{t-1} and Q^t. In terms of the model, this can be expressed as $p_{11} = 1$, $p_{12} = 2$ AR terms, $k_{11} = 0$ and $k_{12} = 1$ when $Q = 1$, and $p_{21} = 0, p_{22} = 1$ AR terms, $k_{21} = 0$ and $k_{22} = 0$ when $Q = 2$. From the model we can see that $p^* \geq 2$, because $p_{im} \leq 2$ for $i = 1, 2$ and $m = 1, 2$.

Some comments on Eq. (5) follow. The set of parents $\mathbf{Pa}_i(X_m)$ of each variable X_m for each state $i \in R(Q)$ is related to a context-specific Bayesian network $\mathcal{B}_i$. Furthermore, depending on that hidden state, each variable X_m may have a different AR order, namely p_{im}, which is upper bounded by p^*. This model must estimate the new parameters $\{\beta_{im0}, ..., \beta_{imk_{im}}, \eta_{im1}, ..., \eta_{imp_{im}}, \sigma_{im}^2\}_{m=1,i=1}^{M,N}$.

Additionally, because the first p^* observations are used as conditionals in Eq. (4), the $\boldsymbol{\pi}$ parameter is shifted to predict the initial distribution of the Q^{p^*} hidden variable, i.e., $\{\pi_i\}_{i=1}^N = \{P(Q^{p^*} = i|\boldsymbol{\lambda})\}_{i=1}^N$. Observe that the complete information probability of an instance $\boldsymbol{x}^{p^*:T}$ of $\boldsymbol{X}^{p^*:T}$ and an instance $\boldsymbol{q}^{p^*:T}$ of $\boldsymbol{Q}^{p^*:T}$ can be expressed as:

$$P(\boldsymbol{q}^{p^*:T}, \boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \lambda) = \pi_{q^{p^*}} \prod_{t=p^*}^{T-1} a_{q^t q^{t+1}} \prod_{t=p^*}^{T} b_{q^t}^{p^*}(\boldsymbol{x}^t).$$
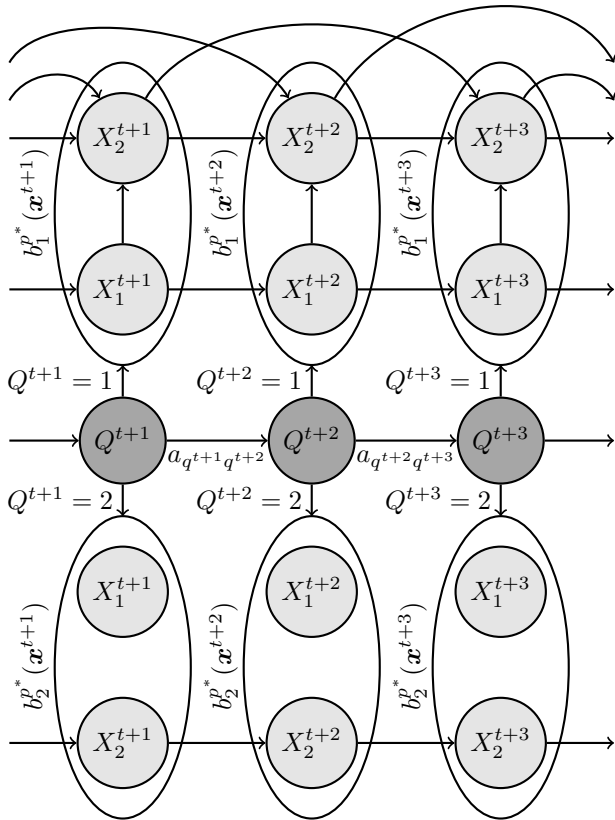

Fig. 2. Graphical representation of an AR-AsLG-HMM model

4.2 Feasibility of the EM algorithm in AR-AsLG-HMMs

To perform the parameter learning, the EM algorithm can be applied. However, we must define an auxiliary function $\mathcal{Q}$ for the log-likelihood defined in Eq. (4). We propose $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$ as the auxiliary function for the EM algorithm, defined as:

$$\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') = \sum_{\boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}') \ln P(\boldsymbol{q}^{p^*:T}, \boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda}).$$ (6)

Moreover, $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$ can be decomposed as:

$$\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') = \sum_{\boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}') \ln P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda})$$
$$+ \ln P(\boldsymbol{x}^{p^*:T}|\boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda}) \sum_{\boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}')$$
$$= \sum_{\boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}') \ln P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}) + LL(\boldsymbol{\lambda})$$
(7)

If we define $\mathcal{H}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$ as the first summand of Eq. (7), i.e.,

$$\mathcal{H}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$$
$$:= \sum_{\boldsymbol{R}(\boldsymbol{Q}^{p^*:T})} P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}') \ln P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda})$$

therefore we have that $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') = \mathcal{H}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') + LL(\boldsymbol{\lambda})$. We now show that if we apply the EM algorithm with $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$, each iteration does not decrease the log-likelihood as required.

Lemma 1. Let $\boldsymbol{\lambda}^{(s)}$ be the parameters at iteration s of the EM and $\boldsymbol{\lambda}^{(s+1)}$ be the resulting parameters after the next iteration of the EM. We have that $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}^{(s+1)}|\boldsymbol{\lambda}^{(s)}) \geq \mathcal{Q}^{p^*}(\boldsymbol{\lambda}^{(s)}|\boldsymbol{\lambda}^{(s)})$.

∎

Lemma 2. Given two arbitrary models with respective parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$, we have that $\mathcal{H}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') \leq \mathcal{H}^{p^*}(\boldsymbol{\lambda}'|\boldsymbol{\lambda}')$, and the equality holds when $P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}) = P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda}')$.

∎

Theorem 1. Let $\boldsymbol{\lambda}^{(s)}$ be the parameters at an iteration s of the EM and $\boldsymbol{\lambda}^{(s+1)}$ be the resulting parameters after the next iteration of the EM. We have that

(a) $LL(\boldsymbol{\lambda}^{(s+1)}) \geq LL(\boldsymbol{\lambda}^{(s)})$. In other words, the log-likelihood of the model cannot worsen after an EM iteration.

(b) The sequence $\{LL(\boldsymbol{\lambda}^{(s)})\}_s$ converges.

∎

The proofs of the lemmas and theorems can be found in the supplementary material.

4.3 The forward-backward algorithm in AR-AsLG-HMMs

As the likelihood function of Eq. (4) and the emission probabilities given by Eq. (5) have changed, the forward-backward algorithm must be adapted. In the E step, we compute the probabilities $\gamma^t(i) = P(Q^t = i|\boldsymbol{x}^{0:T}, \boldsymbol{\lambda})$ for $t = 0, ..., T$ and $i = 1, ..., N$ as the initial point to fit

the forward-backward algorithm. Note that $\gamma^t(i)$ can be expressed as:

$$
\begin{aligned}
\gamma^t(i) &= \frac{P(Q^t = i, \boldsymbol{x}^{p^*:T} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})}{P(\boldsymbol{x}^{p^*:T} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})} \\
&= \frac{P(Q^t = i, \boldsymbol{x}^{p^*:t}, \boldsymbol{x}^{t+1:T} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})}{P(\boldsymbol{x}^{p^*:T} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})} \\
&= \frac{P(\boldsymbol{x}^{t+1:T} | Q^t = i, \boldsymbol{x}^{0:t}, \boldsymbol{\lambda}) P(Q^t = i, \boldsymbol{x}^{p^*:t} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})}{P(\boldsymbol{x}^{p^*:T} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})} \\
&= \frac{\beta_{p^*}^t(i) \alpha_{p^*}^t(i)}{\sum_{j=1}^N \beta_{p^*}^t(j) \alpha_{p^*}^t(j)}
\end{aligned}
\tag{8}
$$

From Eq. (8), the forward variable is $\alpha_{p^*}^t(i) := P(Q^t = i, \boldsymbol{x}^{p^*:t} | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})$ and the backward variable is $\beta_{p^*}^t(i) := P(\boldsymbol{x}^{t+1:T} | Q^t = i, \boldsymbol{x}^{0:t}, \boldsymbol{\lambda})$. Observe that these equations only make sense when $t \geq p^*$. The next lemma shows that we can easily adapt the forward-backward algorithm to compute the α_{p^*} and β_{p^*} parameters of an AR-AsLG-HMM.

Lemma 3. $\alpha_{p^*}^t(i)$ and $\beta_{p^*}^t(i)$ can be computed as:

$$
\alpha_{p^*}^t(i) = \sum_{j=1}^N b_i^{p^*}(\boldsymbol{x}^t) a_{ji} \alpha^{t-1}(j)
$$

$$
\beta_{p^*}^t(i) = \sum_{j=1}^N \beta^{t+1}(j) b_j^{p^*}(\boldsymbol{x}^{t+1}) a_{ij}
\tag{9}
$$

for $t = p^*, ..., T$ and $i = 1, ..., N$, with initial values $\alpha_{p^*}^{p^*}(i) = \pi_i b_i^{p^*}(\boldsymbol{x}^{p^*})$ and $\beta_{p^*}^T(i) = 1$, $i = 1, ..., N$. ∎

4.4 Parameter learning in AR-AsLG-HMMs

To execute the EM algorithm, we must iterate the E step and the M step. For the E step we can use the adapted forward-backward algorithm of Section 4.3 to compute $\gamma^t(i)$ and $\xi^t(i, j)$ for $i, j = 1, ..., N$ and $t = 0, ..., T$:

$$
\gamma^t(i) = \frac{\beta_{p^*}^t(i) \alpha_{p^*}^t(i)}{\sum_{j=1}^N \beta_{p^*}^t(j) \alpha_{p^*}^t(j)}
$$

$$
\xi^t(i, j) = \frac{\alpha_{p^*}^t(i) a_{ij} b_j^{p^*}(\boldsymbol{x}^{t+1}) \beta_{p^*}^{t+1}(j)}{\sum_{u,v=1}^N \alpha_{p^*}^t(u) a_{uv} b_v^{p^*}(\boldsymbol{x}^{t+1}) \beta_{p^*}^{t+1}(v)}
\tag{10}
$$

Computing these quantities is enough for the E step because $\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}')$ can be expressed as:

$$
\mathcal{Q}^{p^*}(\boldsymbol{\lambda}|\boldsymbol{\lambda}') = \sum_{i=1}^N \gamma^{p^*}(i) \ln \pi_i
$$
$$
+ \sum_{t=p^*}^{T-1} \sum_{i=1}^N \sum_{j=1}^N \xi^t(i, j) \ln a_{ij} + \sum_{t=p^*}^T \sum_{i=1}^N \gamma^t(i) \ln b_i^{p^*}(\boldsymbol{x}^t).
\tag{11}
$$

Now, for the M step, we must find the updating formulas for the parameters $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, where $\mathbf{B}$ includes the parameters η_{imr}, β_{imk} and σ_{im}^2. In the following theorem, we provide the updating formulas for the proposed model.

Theorem 2. The M-step for an AR-AsLG-HMM model can be performed using the following updating formulas: parameter $\boldsymbol{\pi} = \{\pi_i\}_{i=0}^N$ is updated as:

$$
\pi_i^* = \gamma^{p^*}(i).
\tag{12}
$$

The parameter $\mathbf{A} = \{a_{ij}\}_{i,j=1}^N$ is updated as:

$$
a_{ij}^* = \frac{\sum_{t=p^*}^{T-1} \xi^t(i, j)}{\sum_{t=p^*}^{T-1} \gamma^t(i)}.
\tag{13}
$$

If we set $f_{im}^t := \boldsymbol{\beta}_{im} \cdot \mathbf{pa}_{im}^t + \boldsymbol{\eta}_{im} \cdot \boldsymbol{d}_{im}^t$, the parameters $\{\eta_{imr}\}_{r=1}^{p_{im}}$, $\{\beta_{imk}\}_{k=0}^{k_{im}}$ can be updated jointly, solving the following linear system:

$$
\begin{cases}
\sum_{t=p^*}^T \gamma^t(i) x_m^t = \sum_{t=p^*}^T \gamma^t(i) f_{im}^t \\
\sum_{t=p^*}^T \gamma^t(i) x_m^t u_{im1}^t = \sum_{t=p^*}^T \gamma^t(i) u_{im1}^t f_{im}^t \\
\quad\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\sum_{t=p^*}^T \gamma^t(i) x_m^t u_{imk_{im}}^t = \sum_{t=p^*}^T \gamma^t(i) u_{imk_{im}}^t f_{im}^t \\
\sum_{t=p^*}^T \gamma^t(i) x_m^t x_m^{t-1} = \sum_{t=p^*}^T \gamma^t(i) x_m^{t-1} f_{im}^t \\
\quad\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\sum_{t=p^*}^T \gamma^t(i) x_m^t x_m^{t-p_{im}} = \sum_{t=p^*}^T \gamma^t(i) x_m^{t-p_{im}} f_{im}^t
\end{cases}
\tag{14}
$$

If we set $\hat{f}_{im}^t := \beta_{im0}^* + \beta_{im1}^* u_{im1}^t + \cdots + \beta_{imk_{im}}^* u_{imk_{im}}^t + \eta_{im1}^* x_m^{t-1} + \cdots + \eta_{imp_{im}}^* x_m^{t-p_{im}}$, then, σ_{im}^2 can be updated as:

$$
(\sigma_{im}^2)^* = \frac{\sum_{t=p^*}^T \gamma^t(i)(x_m^t - \hat{f}_{im}^t)^2}{\sum_{t=p^*}^T \gamma^t(i)}.
\tag{15}
$$

This update must be done for every variable $m = 1, ..., M$ and hidden state $i = 1, ..., N$. ∎

Eq. (14) forms a linear system of $k_{im} + p_{im} + 1$ unknowns with $k_{im} + p_{im} + 1$ equations. If the resulting context-specific Bayesian model for every hidden state is a naïve Bayesian network and $p_{im} = 0$ for $i = 1, ..., N$ and $m = 1, ..., M$, then we only require to update the parameters $\{\beta_{im0}\}_{i=1,m=1}^{N,M}$. Its updating formula is:

$$
\beta_{im0}^* = \frac{\sum_{t=p^*}^T \gamma^t(i) x_m^t}{\sum_{t=p^*}^T \gamma^t(i)}
$$

Otherwise, we must solve the linear system which can be done using exact or iterative methods. If we use for example the Gauss-Jordan reduction algorithm to solve the linear system, an additional computational cost of $O((k_{im} + p_{im} + 1)^3)$ must be assumed. Therefore, simpler structures are recommended in order to not slow down the learning process. This requirement is taken into account during the SEM algorithm as mentioned in Section 3.5.

A pseudocode of the adapted EM algorithm can be found in Fig. 3.

4.5 The Viterbi algorithm in AR-AsLG-HMMs

In the following lemma we show that the traditional Viterbi algorithm can be adapted to determine the most probable sequence of hidden states in AR-AsLG-HMMs.

Lemma 4. If $\delta_{p^*}^t(i) = \max_{\boldsymbol{q}^{p^*:t-1}} \{P(\boldsymbol{x}^{p^*:t}, \boldsymbol{q}^{p^*:t-1}, Q^t = i | \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})\}$ represents the most probable sequence of hidden states up to time $t - 1$ for state i at time t, then $\delta_{p^*}^t(i)$ can be computed recursively.

$$
\delta_{p^*}^t(i) = \max_{j=1,...,N} \{\delta_{p^*}^{t-1}(j) a_{ji}\} b_i^{p^*}(\boldsymbol{x}^t)
$$

Input: A prior parameter $\boldsymbol{\lambda}^{(0)}$, a dataset $\boldsymbol{X}^{0:T}$
Output: A learned parameter $\boldsymbol{\lambda}^*$
1: **for** $s = 0, 1, ...$ until convergence in likelihood is met: **do**
2:    Use Eq. (9) to perform the adapted forward-backward algorithm with $\boldsymbol{\lambda}^{(s)}$
3:    Estimate $\gamma^t(i)$ and $\xi^t(i, j)$ with Eq. (10)
4:    Use Eq. (12) and Eq. (13) to obtain: $\boldsymbol{\pi}^*$ and $\mathbf{A}^*$
5:    **for** $i = 1, ..., N$ and $m = 1, ..., M$ **do**
6:       Use Eq. (14) to obtain:
7:       $\{\beta^*_{im0}, \beta^*_{im1}, ..., \beta^*_{imk_{im}}, \eta^*_{im1}, ..., \eta^*_{imp_{im}}\}$
8:       Use Eq. (15) to obtain: $(\sigma^2_{im})^*$
9:    **end for**
10: **end for**
11: **return** $\boldsymbol{\lambda}^*$

Fig. 3. Pseudocode for the adapted EM algorithm

The Viterbi algorithm is initialized with $\delta^{p^*}_{p^*}(i) = \pi_i b^{p^*}_i(\boldsymbol{x}^{p^*})$.

∎

4.6 The SEM algorithm in AR-AsLG-HMMs

Regarding the structural optimization process, the SEM algorithm for AR-AsLG-HMMs must also be modified. The proposed auxiliary function is:

$$\mathcal{Q}^{p^*}(\mathcal{B}, \boldsymbol{\lambda}|\mathcal{B}', \boldsymbol{\lambda}') = E_{P(\boldsymbol{q}^{p^*:T}|\boldsymbol{x}^{0:T}, \mathcal{B}', \boldsymbol{\lambda}')}$$
$$\times [\ln P(\boldsymbol{x}^{p^*:T}, \boldsymbol{Q}^{p^*:T}|\mathcal{B}, \boldsymbol{x}^{0:p^*-1}, \boldsymbol{\lambda})] - 0.5\#(\mathcal{B})\ln(T). \quad (16)$$

The steps for the adapted SEM algorithm are the same as in the general SEM. However, we must consider that given a time slice t, the algorithm must not only look for the best instantaneous structure at time t or the best structure with variables $(X^t_1, ..., X^t_M)$ but also look for the best transition structure at time t or the relationships between $(X^t_1, ..., X^t_M)$ variables and their AR versions, i.e., $(X^{t-1}_1, X^{t-2}_1, ..., X^{t-p^*-1}_M, X^{t-p^*}_M)$, which implies that the search space dimension increases. More specifically, we have to search not only in the space of directed acyclic graphs (DAGs) for the best instantaneous structures, but also in the space $S_{p^*} = \{0, 1, ..., p^*\}^N \times \{0, 1, ..., p^*\}^M$, for the best transition structure. A component p_{im} of a matrix $\boldsymbol{p} \in S_{p^*}$ indicates the number of lags for variable X_m in the hidden state $i \in R(Q)$. For instance, if $p_{im} = 2$, X^t_m has incoming arcs from the variables X^{t-2}_m and X^{t-1}_m when $Q^t = i$. A pseudocode of the adapted SEM is given in Fig. 4.

It is pertinent to mention that in the SEM algorithm, in the step of finding $\mathcal{B}^{(s)} = \arg\max_{\mathcal{B}} \mathcal{Q}^{p^*}(\mathcal{B}, \boldsymbol{\lambda}^{(s)}|\mathcal{B}^{(s-1)}, \boldsymbol{\lambda}^{(s)})$ it is not necessary to use Eq. (16), since the initial distribution and the transition matrix are kept unchanged. We can take advantage of the linearity of Eq. (16) to compare structures, i.e., if a dependency of X_m has been added or deleted (AR or parent parameter) at the hidden state i, it is reasonable to use the following score:

$$\text{score}_{im} = \sum_{t=p^*}^{T} \gamma^t(i) \ln(\mathcal{N}(x^t_m|f^t_{im}, \sigma^2_{im})). \quad (17)$$

Input: A prior parameter $\boldsymbol{\lambda}^{(0)}$, a dataset $\boldsymbol{X}^{0:T}$, a prior structure $\mathcal{B}^{(0)}$
Output: A learned parameter $\boldsymbol{\lambda}^*$ and structure $\mathcal{B}^*$
1: Solve $\boldsymbol{\lambda}^{(1)} = \arg\max_{\boldsymbol{\lambda}} \mathcal{Q}^{p^*}(\mathcal{B}^{(0)}, \boldsymbol{\lambda}|\mathcal{B}^{(0)}, \boldsymbol{\lambda}^{(0)})$ with the EM algorithm
2: **for** $s = 1, ...$ until convergence in the penalized log-likelihood is met **do**
3:    Solve $\mathcal{B}^{(s)} = \arg\max_{\mathcal{B}} \mathcal{Q}^{p^*}(\mathcal{B}, \boldsymbol{\lambda}^{(s)}|\mathcal{B}^{(s-1)}, \boldsymbol{\lambda}^{(s)})$ with any meta-heuristic method   ▷ Here we obtain a DAG for the instantaneous time structure, and a vector $\boldsymbol{p}$ for the transition time structure
4:    Solve $\boldsymbol{\lambda}^{(s+1)} = \arg\max_{\boldsymbol{\lambda}} \mathcal{Q}^{p^*}(\mathcal{B}^{(s)}, \boldsymbol{\lambda}|\mathcal{B}^{(s-1)}, \boldsymbol{\lambda}^{(s)})$ with the EM algorithm
5: **end for**
6: **return** $\boldsymbol{\lambda}^*$ and $\mathcal{B}^*$

Fig. 4. Pseudo-code for the adapted SEM algorithm

If changes have been done to many variables in many hidden states, it is better to use the following score:

$$\text{score} = \sum_{i=1}^{N} \sum_{m=1}^{M} \text{score}_{im}. \quad (18)$$

To perform the structural optimization step, we must search in the space of structures. In this article we use a heuristic forward greedy algorithm to perform the structure optimization. In this approach, we initialize all the structures in a naïve form with no AR parameters. During the optimization, we visit each variable for each hidden state and add AR or parent dependencies as long as Eq. (17) improves. Its pseudocode is shown in Fig. 5.

Other algorithms have been used to search in the graph space during the SEM algorithm, e.g., [5] used a tabu search algorithm [34], and [27] used a simulated annealing algorithm [35]. In general, any meta-heuristic or heuristic can be used to search in the space of graphs.

4.7 Hidden states labelling

In practice, when HMMs are used, categorical labels are given to the hidden states for interpretation purposes. However, only after training the model, the model parameters are manually checked to determine which categorical label corresponds with each trained hidden state. Here, we propose an automatic numerical labelling for trained models, where a numerical function is used to label a trained hidden state. Let $g : R(Q) \rightarrow \mathcal{R}$ be a function that maps each hidden state into a real number depending on the models parameters. This function g not only helps us determine whether a change in hidden states occurs but also the magnitude of the change. For example, if deviations from known standards or desired values $\boldsymbol{\kappa} = \{\kappa_1, ..., \kappa_m\}$ of $\boldsymbol{X}$ imply changes in state, the following g functions described in Eq. (19) and Eq. (20) can be used to help in hidden states labelling in AR-AsLG-HMMs:

$$g_1(i) = \sum_{m=1}^{M} v_m(\nu_{im} - \kappa_m) \quad (19)$$

$$g_2(i) = \max_{m=1,...,M} \{v_m(\nu_{im} - \kappa_m)\} \quad (20)$$

Input: A parameter $\boldsymbol{\lambda}^{(s)}$, a prior structure $\mathcal{B}^{(s-1)}$
Output: A structure $\mathcal{B}^{(s)}$

```
 1: for i = 1, ..., N do          ▷ Optimization of AR structures
 2:     for m = 1, ..., M do
 3:         Compute score_im with B^(s-1) and λ^(s)
 4:         while p_ik ≤ p* do
 5:             Define B̂ with p̂_ik := p_ik + 1 and estimate its
    parameters λ̂ = arg max_λ Q^{p*}(B̂, λ|B^(s-1), λ^(s))
 6:             Compute scorê_im with B̂ and λ̂
 7:             if scorê_im > score_im then
 8:                 Update B^(s-1), score_im and λ^(s)
 9:             else Break
10:             end if
11:         end while
12:     end for
13: end for
14: for i = 1, ..., N do ▷ Optimization of non-AR structures
15:     for m = 1, ..., M do
16:         Compute B_im := all the possible DAG graphs
    resulting by adding one arc to B^(s-1) in its i context-
    specific Bayesian network, where X_m is a new children
    of any variable
17:         if B_im is non-empty then
18:             Compute score_im with B^(s-1) and λ^(s)
19:             for B̂ in B_im do
20:                 Estimate the parameters λ̂ =
    arg max_λ Q^{p*}(B̂, λ|B^(s-1), λ^(s))
21:                 Compute scorê_im with B̂ and λ̂
22:                 if scorê_im > score_im then
23:                     Update B^(s-1), score_im and λ^(s)
24:                 end if
25:             end for
26:         end if
27:     end for
28: end for
29: Set B^(s) := B^(s-1)
```

Fig. 5. Pseudo-code for the forward greedy algorithm

Where

$$\nu_{im} = \frac{\beta_{im0} + \beta_{im1}\nu_{iu_{im1}} + \cdots + \beta_{imk_{im}}\nu_{iu_{imk_{im}}}}{1 - (\eta_{im1} + \cdots + \eta_{imp_{im}})}. \quad (21)$$

Observe that in Eq. (21), the value of ν_{im} depends on the ν value of the parents of variable X_m in the context-specific graph related to the i state, so Eq. (19) and Eq. (20) must be calculated recursively. The recursion begins with those variables that fulfil the following condition: $\mathbf{Pa}_i(X_m) = \emptyset$. Next, the recursion is computed for their descendants in the context-specific graph, until no variables are left. In general, ν_{im} can be interpreted as the mean of variable X_m at the hidden state i. Additionally, the vector $\mathbf{v} = (v_1, ..., v_M)$ for $m = 1, .., M$ can be considered a feature relevance constant vector or a scaling constant vector that can be tuned according to the nature of the problem.

Eq. (19) can be used in cases where the addition of errors determines the driven process. For example, in the case of a country economy where the aggregation of economic variables can determine if there is economic growth or not. Or in the case of bearings degradation, where the aggregation of

the amplitude of desired frequencies represents the presence of failure. On the other hand, Eq. (20) can be used when high deviations from a single variable is enough to determine the dynamical process. For example, consider a patient with a chronic disease with many sensors that measure different biological variables. For each variable there is a desirable value that determine good health. If only one variable drifts from the desirable value, the health of the patient can be in danger. In conclusion, the experiment and context of the problem may require a different g function to describe the hidden states.

5 EXPERIMENTS

In this section, we will compare our model (AR-AsLG-HMMs) with AsLG-HMMs, LMSAR, AR-MoG-HMMs, MoG-HMMs, VAR-MVGHMMs, BMMs and a simple AR-AsLG-HMM with naïve Bayes context-specific Bayesian networks that we will call naïve-HMMs (this kind of models have been used in [36]). In the case of LMSAR [8][3] and AR-MoG-HMM [10], it was defined only for one variable. Therefore, in these experiments, we will assume that for these models every variable is independent. In particular, LMSAR is a special case of an AR-AsLG-HMM, where only AR parameters are used in the mean, but the number of them do not change with the hidden state and the variance of the model does not depend on the hidden state. In the case of AR-MoG-HMM, the model assumes that the variances are unitary and do not depend on the hidden state; in spite of that it cannot be expressed directly as an AR-AsLG-HMM. Also, both AR-AsLG-HMM and AsLG-HMM use the forward-greedy algorithm in the SEM algorithm to ensure reproducibility. The aim is to show the capabilities of our model to change the number of AR parameters and the context-specific Bayesian networks when they are needed.

Experiments with synthetic data are performed. The data are generated such that over time, the AR process changes. Two dynamic processes are used with six variables. The models are learned using only one time series, where three possible hidden states are present and appear in time blocks. We aim to determine for new data the most likely sequence of hidden states. This sequence tells us the current probabilistic distribution of the data and therefore which probabilistic relationships are relevant. Also, the number of parameters and BIC score play an important role to identify which model is better as explained in Section 4.6.

Air quality data and real bearing degradation data are also used. The p^* values are computed using the Yule-Walker equations. These are used as well for AR-MoG-HMM, BMM, VAR-MVGHMM and LMSAR to determine the maximum lag during the learning task. For the mixture models, two and three mixture components were used, and the models with two mixture components had the highest BIC and log-likelihood. Then, just two mixture components were used.

For both synthetic and real data, the models are initialized with a uniform transition matrix $\mathbf{A}$, specifically $a_{ij} = 1/N$; for $i, j = 1, ..., N$; the same for the initial distribution $\boldsymbol{\pi}$, specifically, $\pi_i = 1/N$ for $i = 1, ..., N$. In
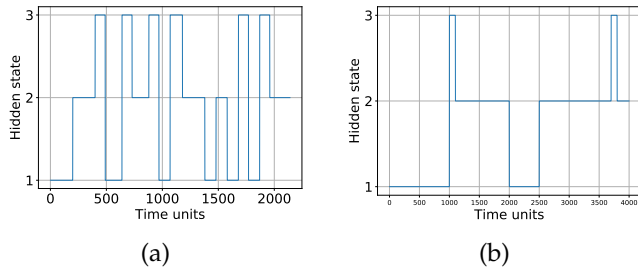
3. See page 57

Fig. 6. Sequences of hidden states used to construct the training signals for scenario 1 (a) and for scenario 2 (b)

Fig. 7. Sequences of hidden states used to construct the test signals. Sequence 1 (a) and Sequence 2 (b) are used for both scenarios

the case of our AR-AsLG-HMM, we evaluate the partial correlation function up to five AR values to prevent high computational times and set $p_{im} = 0$ for $i = 1, ..., N$ and $m = 1, ..., M$; this means that no AR relationships are assumed a priori in the models. For both AR-AsLG-HMM and AsLG-HMM, all the context-specific Bayesian networks are initialized as naïve Bayes networks. The emission probabilities for the AR-AsLG-HMM and AsLG-HMM are initialized with $\beta_{im0} = i(\max_t x_m^t - \min_t x_m^t)/(N+1) + \min_t x_m^t$ and $\sigma_{im}^2 = 2(\max_t x_m^t - \min_t x_m^t)$ for $i = 1, ..N$ and $m = 1, ..., M$. The purpose of this selection for β_{im0} is to initialize the mean of each variable for each hidden state in an equally separated different point in the possible range of values given by the training dataset. The selection of σ_{im}^2, is to avoid infinite or nan values in the first iterations of the forward-backward algorithm. For the mixture models, the distribution of the mixture coefficients is uniform, and the mean coefficients for the mixtures models are initialized using a k-means algorithm of clustering.

All the models were implemented in python 3 and the used libraries were numpy, matplotlib, pandas, networkx, math, pickle and the k-means algorithm from the scikit-learn library. No parallelization or own created external functions or libraries (like in C or C++) to improve the performance were used. The software in python 3 will be available upon request after publication.

5.1 Synthetic data

We consider two scenarios with three known hidden states. One follows AR-AsLG-HMM emission probabilities and another AR-MoG-HMM emission probabilities. We generate blocks of data for each hidden state. We mix these blocks as indicated by Fig. 6 depending on the scenario to create a signal and train every model with it. We try to simulate data as in real life applications where hidden states may not have a particular order of appearance i.e., any possible transition between hidden states is possible. We will evaluate the learned models with two different types of sequences of hidden states. These two sequences are generated fifty times to be evaluated in the testing phase. From the fifty evaluations we report the the mean log-likelihood (LL), mean BIC, the standard deviation of the log-likelihoods (Std) and the number of parameters in the model (#).

Both scenarios use six variables. From the parameters, in the case of AR-AsLG-HMM emission probabilities, we have a hidden state with no structural complexity (no-AR and no-parent relationships in f_{im}^t), a second one with some
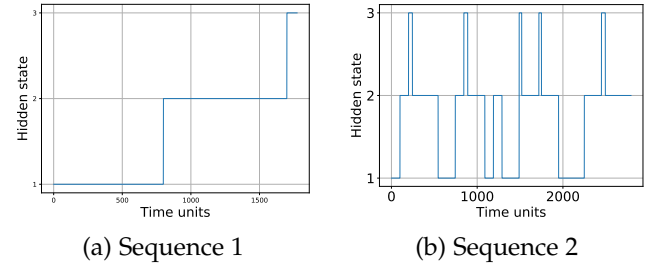
structural complexity and the last one with a complex structure (several AR and parent relationships in f_{im}^t). We also edit the parameters in AR-AsLG-HMM in such a way that the more complex the context-specific Bayesian networks are, the greater amplitudes for the $g_1(i)$ (Eq. (19)) function are. The parameters used for the hidden states for both AR-AsLG-HMM and AR-MoG-HMM can be found in the supplementary material. The $g_1(i)$ function used for all the experiments has $v_m = 1$ for $m = 1, ..., M$ and $\kappa_m = 0$, for $m = 1, ..., M$. The sequence of hidden states used to construct the training signal for both scenarios can be seen in Fig. 6. The two sequences of hidden states used to generate the testing signals (fifty testing signals are generated for each sequence and each scenario) are illustrated in Fig. 7.

Seq	Model	mean LL	mean BIC	Std	#
1	AR-AsLG-HMM	**-25909.45**	**52432.48**	86.19	64
	AsLG-HMM	-32817.77	66181.78	193.74	55
	LMSAR	-30389.47	61587.06	112.36	108
	AR-MoG-HMM	-28960.25	59357.17	25.02	192
	MoG-HMM	-68411.13	138124.24	**1.67**	174
	Naïve-HMM	-33251.80	66997.46	199.35	**48**
	BMM	-56348.00	113997.98	19.58	174
	VAR-MVGHMM	-68243.34	140415.08	0.84	525
2	AR-AsLG-HMM	**-41478.86**	**83608.50**	87.76	64
	AsLG-HMM	-54167.33	108914.01	176.69	55
	LMSAR	-48356.20	97569.52	94.83	108
	AR-MoG-HMM	-45547.16	92618.09	32.21	192
	MoG-HMM	-107682.31	216745.54	**1.72**	174
	Naïve-HMM	-54395.72	109315.24	163.32	**48**
	BMM	-88693.42	178767.76	21.12	174
	VAR-MVGHMM	-107504.80	219176.15	1.50	525

TABLE 3
Results for each testing sequence of scenario 1

Seq	Model	mean LL	mean BIC	Std	#
1	AR-AsLG-HMM	**-19822.87**	**40573.45**	114.66	106
	AsLG-HMM	-20255.87	41304.78	118.92	88
	LMSAR	-26212.61	53233.23	112.16	108
	AR-MoG-HMM	-22990.46	47417.38	10.42	192
	MoG-HMM	-52213.44	105728.67	2.17	174
	Naïve-HMM	-23135.52	46764.83	138.46	**48**
	BMM	-40292.46	81886.72	23.05	174
	VAR-MVGHMM	-52069.19	108066.19	**0.76**	525
2	AR-AsLG-HMM	**-32883.36**	**66750.73**	192.35	106
	AsLG-HMM	-33867.52	68576.21	236.21	88
	LMSAR	-44482.64	89822.32	226.24	108
	AR-MoG-HMM	-36504.69	74533.01	14.41	192
	MoG-HMM	-82248.49	165877.78	3.95	174
	Naïve-HMM	-38804.18	78132.11	241.70	**48**
	BMM	-63655.48	128691.74	39.42	174
	VAR-MVGHMM	-82064.02	168294.22	**1.04**	525

TABLE 4
Scores for each testing sequence of scenario 2

Model	Times 1 (s)	Times 2 (s)
AR-AsLG-HMM	6.842	55.098
AsLG-HMM	4.608	33.009
LMSAR	70.797	2.458
AR-MoG-HMM	189.114	223.762
MoG-HMM	110.749	190.766
Naïve-HMM	3.702	8.904
BMM	266.231	5762.679
VAR-MVGHMM	11.165	131.059

TABLE 5
Scenario 1 and 2 learning times

In Table 3 and Table 4, we observe the results for scenario 1 and 2 respectively, for both sequences. We observe that AR-AsLG-HMM obtained the best results in LL and BIC score. The naïve-HMM, AsLG-HMM and AR-MoG-HMM obtained fair results. The mixture models: MoG-HMM, VAR-MVGHMM and BMM obtained poor results in LL and BIC score. In the case of BIC score, the penalization for mixture models was higher since a greater number of parameters were needed for these models. In terms of standard deviation, MoG-HMM and VAR-MVGHMM obtained the best results, nevertheless also they obtained the worst results in BIC score. Next, we find that AR-MoG-HMM, AR-AsLG-HMM and LMSAR obtained fair results in standard deviation with a good BIC score. Finally, AsLG-HMM and naïve-HMM obtained the worst standard deviation values in spite of their BIC score. In terms of the number of parameters, the naïve-HMM used the fewest number of parameters since it has the simplest structure and VAR-MVGHMM used the highest number of parameters since it considers cross-AR dependencies between variables. AR-AsLG-HMM and AsLG-HMM used fewer parameters than mixture models, whereas mixture models used between four to eleven times the number of parameters used by naïve-HMM. We also observe that LMSAR had a fair number of parameters since it assumes independence between variables for all hidden states, standard deviations independent of the hidden state and only AR parameters are used. In the supplementary material an analysis of the obtained Viterbi paths for some sequences can be found.

For training the models we set the maximum number of EM iterations to 200 and the convergence threshold to 1×10^{-10}. For the SEM, the number of iterations are set to 200 and the convergence threshold to 1×10^{-5}. Table 5 shows the required times for learning the models in each scenario. All the models converged with some exceptions. In scenario 1, MoG-HMM and VAR-MVGHMM had to limit their number of EM iterations since singular-covariance matrices raised in the parameters, in particular, MoG-HMM had to iterate 8 times and VAR-MVGHMM iterated 26 times before singular covariance matrices appeared. In the case of scenario 2, LMSAR and MoG-HMM had similar problems and MoG-HMM iterated 13 times and LMSAR just could iterate 1 time. We can observe that BMM is the most expensive in time among all the models. This is due to the structure learning process that it does [23], where several mutual information quantities must be computed to determine the best AR-relationships. On the other hand, the fastest algorithm that converged was the naïve-HMM, which was expected since it had the simplest structure of all the models. In spite of that, we observe that AR-AsLG-HMM and AsLG-HMM obtained the second best times for training, and the remaining models had longer training times.

We can observe from these experiments that AR-AsLG-HMM is capable of being simple enough to explain linear Gaussian autoregressive and mixture Gaussian processes and prevent overfitting, but can be complex enough to detect relevant parameters that drive the hidden states. AsLG-HMM has this property as well, but as can be seen from the obtained BIC scores and standard deviations, the AR variables are pertinent. In terms of variance of the predictions, AR-AsLG-HMM had decent results, which implies it is stable.

5.2 Real data

5.2.1 Air quality in Beijing

Here we use a dataset found in the UCI Machine Learning Repository named: "Beijing Multi-Site Air-Quality Data Data Set" [37]. The dataset consists of measurements of air quality in different monitoring stations in Beijing. We in particular take the measurements from the file "PRSA Data Aotizhongxin" which represent the name of the monitoring station Aotizhongxin. This dataset has hourly measurements from March 2013 until February 2017. The data contains missing data (3.37% of the dataset for the selected variables). The missing data is filled using the mean of the values of the five previous hours. The hidden variable in this problem can be understood as the air quality. For this study we use the following variables: sulfur dioxide (SO$_2$ in $\mu g/m^3$), nitrogen dioxide (NO$_2$ in $\mu g/m^3$), carbon monoxide (CO in $\mu g/m^3$), ozone (O$_3$ in $\mu g/m^3$), coarse particulate matter (PM$_{10}$ in $\mu g/m^3$) and fine particulate matter (PM$_{2.5}$ in $\mu g/m^3$). Bayesian networks and HMM have been used before to determine air quality [38], [39], [40], [41], showing advantages in the generation of information and discovery of relationships between variables.

The Chinese air quality limits for hourly, daily and monthly measurements are expressed in the law GB 3095-2012. These limits are used to model the g function for this problem. In particular, $\boldsymbol{\kappa} = \{500, 200, 10000, 200, 150, 75\}$ and $\mathbf{v} = \{1/500, 1/200, 1/10000, 1/200, 1/150, 1/75\}$. The g function in this case uses Eq. (20). If $g_2(i) > 0$ it means that one or many variables are above the permissible limit and the air quality is pretty bad. Great negative values are desirable for g_2 since it implies good air quality. The aim is to learn models to determine the air quality when new observations arrive. We use the first year of data to train the models: from march of 2013 to February of 2014. A first experiment with only two hidden states is considered with $p^* = 1$. We use this model to check if the model is capable of determining in a binary manner the air quality using AR processes of order one. Later, another model is trained where the number of hidden states is set using the naïve-HMM since this model is the simplest one. The selection of the number of hidden states could be done with the same AR-AsLG-HMM but for fairness, this strategy is used. Two to eleven hidden states were considered, but with three hidden states, naïve-HMM obtained the best LL for the year 2013 and all the models could be trained. For each model

Model	mean LL	mean BIC	Std	#
AR-AsLG-HMM	-167390.66	335516.71	1646.68	69
AsLG-HMM	-221071.33	442778.17	3183.77	58
LMSAR	-183257.57	366841.98	3034.64	36
AR-MoG-HMM	**-138174.57**	**277493.09**	**405.75**	126
MoG-HMM	-213999.24	429033.48	2692.62	114
Naïve-HMM	-228682.23	457745.78	2126.03	**30**
BMM	-214429.71	429894.41	2521.67	114
VAR-MVGHMM	-162813.92	326826.25	1393.19	132

TABLE 6
Air quality scores when two hidden states are used

Model	mean LL	mean BIC	Std	#
AR-AsLG-HMM	-161077.29	323525.48	3346.94	133
AsLG-HMM	-214970.27	430930.13	3666.78	91
LMSAR	-186458.75	373898.02	3702.09	108
AR-MoG-HMM	**-138001.78**	**277746.70**	**210.11**	192
MoG-HMM	-211794.34	425168.40	5376.60	174
Naïve-HMM	-219751.80	440102.81	2570.88	**48**
BMM	-211524.89	424629.51	4456.12	174
VAR-MVGHMM	-155614.62	315995.65	2875.60	525

TABLE 7
Air quality scores when three hidden states are used

we predict individually the air quality of the three following years of data: from March of 2014 to February of 2017. As above, we record the mean likelihoods, mean BICs, standard deviation of likelihoods and number of parameters of each model.

Table 6 and Table 7 show the scores obtained. We can observe that AR-AsLG-HMM, VAR-MVGHMM and AR-MoG-HMM attained the best results in the LL and BIC scores. The remaining models got fair results. In terms of stability, we see that AR-MoG-HMM has the lowest standard deviation, followed by BMM, naïve-HMM and AR-AsLG-HMM. In terms of the number of parameters, we observe that naïve-HMM and AsLG-HMM have the fewest number of parameters. Followed by these models, LMSAR and AR-AsLG-HMM achieved a fair number of parameters and finally, mixture models, as expected, had to use a great amount of parameters.

Fig. 8 shows the predicted air quality for the first two weeks of 2016 for each model using the Viterbi algorithm when two hidden states are used. Real readings are shown in Fig. 8(i), where we express 1 when any of the variables surpasses the law limits and -1 when all the variables are under the law limits. From Fig. 8(i), we observe that there are four periods of time where pollution levels out of the legal levels are found: from 0 to 75 hours, from 115 to 120 hours, from 186 to 219 hours and from 322 to 360 hours. Clearly Fig. 8(i) does not tell us the severity of the pollution nor the closeness to an outlaw pollution level. We can see that the model with the highest score (AR-MoG-HMM Fig. 8(d)) shows a horizontal line below zero, which implies that the pollution level is always close to an outlaw level, which is not consistent with what is shown in Fig. 8(i). The next model with the highest LL and BIC is VAR-MVGHMM Fig. 8(h), which shows a noisy behaviour but always above zero, indicating a persistent outlaw pollution level with changes in severity; however it does not match with the reality observed in Fig. 8(i). In the case of LMSAR, we observe transitions between legal and illegal pollution levels; however, it reads as persistent high pollution levels which

(a) AR-AsLG-HMM

(b) AsLG-HMM

(c) LMSAR

(d) AR-MoG-HMM

(e) MoG-HMM

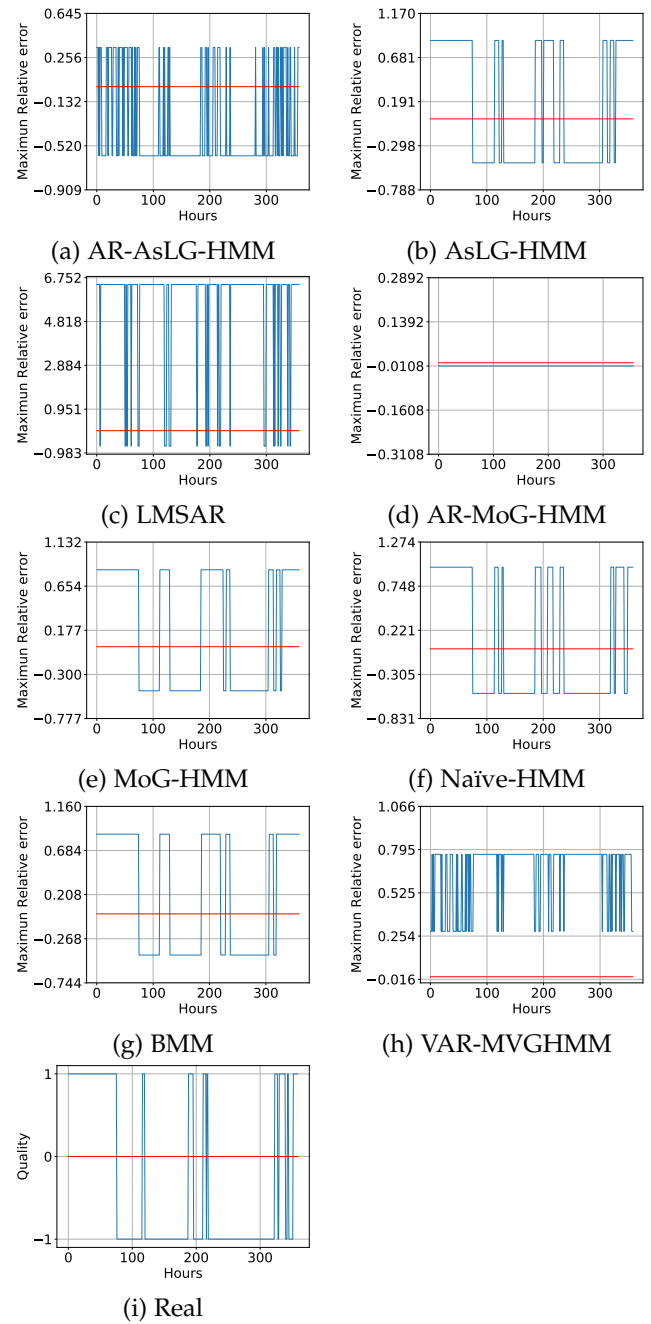(f) Naïve-HMM

(g) BMM

(h) VAR-MVGHMM

(i) Real

Fig. 8. Viterbi paths for the air quality example during the first week of 2016 when two hidden states are used

are not consistent with Fig. 8(i). AR-AsLG-HMM shows a noisy prediction; however, in this case there are variations between outlaw levels and legal levels of pollution. There are four moments where the pollution levels are illegal in Fig. 8(a) as in Fig. 8(i); however, the prediction is not so clear as in the case of AsLG-HMM Fig. 8(b), MoG-HMM Fig. 8(e), naïve-HMM Fig. 8(f) and BMM Fig. 8(g), where more consistent predictions are found with similar levels of the $g_2(i)$ function. Since the Viterbi paths achieved for three hidden states are similar to those with two hidden states, they are shown in the supplementary material.

The noisy predictions can be explained using the learned transition matrices: in the case of non-AR models, the transition probabilities were concentrated on the diagonal of the
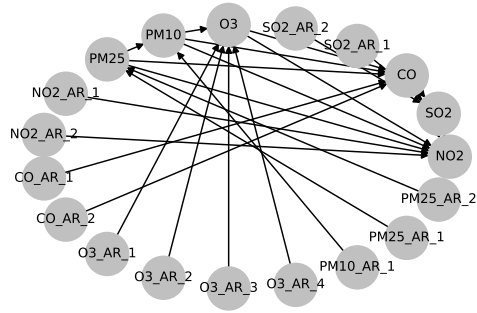
(a) $g(i) = -0.78$

(b) $g(i) = 0.7$

Fig. 9. Context-specific graphs learned by AR-AsLG-HMM. (a) shows a graph where the air quality is good and (b), where the air quality is bad.

transition matrix as in the case of AsLG-HMM in Eq. (22) with matrix $\boldsymbol{A_1}$; whereas AR models learned more uniform transition matrices as in the case of AR-AsLG-HMM in Eq. (22) with matrix $\boldsymbol{A_2}$.

$$\boldsymbol{A_1} = \begin{bmatrix} 0.96 & 0.04 \\ 0.04 & 0.96 \end{bmatrix}, \boldsymbol{A_2} = \begin{bmatrix} 0.80 & 0.20 \\ 0.38 & 0.62 \end{bmatrix} \quad (22)$$

The latter causes more likely jumps between hidden states and noisy Viterbi paths can be obtained. Nonetheless, from all the AR models, AR-AsLG-HMM was the only one closest to the real scenario given by Fig. 8(i).

Fig. 9 shows two learned graphs when three hidden states were assumed. In the context-specific Bayesian networks, AR variables are denoted as $X_m\_AR\_r$, where r is the number of lags for the variable X_m. In Fig. 9(a) we show a graph when the air quality is good and in Fig. 9(b) it is bad. In both graphs we can observe some interesting relationships similar to the ones found in [38]. For example, in Fig. 9(a) we observe that CO depends on $PM_{2.5}$ and PM_{10} and SO_2 and NO_2 are related to CO. These relationships come from the process of combustion of gas and charcoal. Also NO_2 is related to O_3 which indicates the photochemistry of NO_2 for the production of O_3. In Fig. 9(b) we see that these relationships remain. However, the dependences on previous values for each variable changes, which tells us the level of impact of the past on the pollution levels.

5.2.2 Ball-bearings degradation

Ball-bearings are used inside many mechanic tools as drills, rotors, etc. Ball-bearings represent critical components inside these machines. The failure or degradation of these components can be translated to loses in time, money and assets for industries. Monitoring ball-bearings is crucial and relevant, and the use of HMM can give insight of the bearing degradation process and therefore help in the development of maintenance policies [42].

The benchmark used to validate the proposed model in this section comes from ball-bearing vibrational data [43]. The run-to-failure tool machine setup is shown in Fig. 11. Four ball-bearings are tested in the setup. The signals are obtained with vibrational sensors. The desired vibrations are submerged in noise; therefore, filtration techniques are required. In this study, the signals are filtered as in [44], where spectral kurtosis algorithms are used. From the filtered signal, we calculate its spectrum with the Fourier transform and the ball-bearing fundamental frequencies, namely, ball pass frequency outer (BPFO) related to the ball-bearings outer race, ball pass frequency inner (BPFI) related to the ball-bearings inner race, ball spin frequency (BSF) related to the ball-bearings rollers and the fundamental train frequency (FTF) related to the ball-bearings cage.

The training signal consists of 2156 records, while the testing signal has 6324 records. We use the fundamental frequencies as variables of the models, hence four variables are used. We must recall that the dataset comes from a coupled mechanical system. Therefore, in the presence of a fail in any part of the system, vibrations will be generated that will transmit across the whole system. Fig. 10 shows the BPFO frequency of every testing ball-bearing. As we can see, for all the ball-bearings, the magnitude of their frequencies grows abruptly at the end of the measures indicating a phase of ball-bearing failing somewhere in the mechanical setup. In the training dataset, Bearing 3 fails due to its inner race and Bearing 4 due to its rollers. In the testing dataset Bearing 3 fails due to its outer race. The hidden variable for this context can be understood as the ball-bearing health state. The number of hidden states was set depending on the scores obtained by naïve-HMM in the training data. We observed that with seven hidden states, the scores of the naïve-HMM were optimized.

For this problem, we constantly obtained underflow problems due to the small amplitudes of some frequencies. Therefore, all the dataset were multiplied by 1000. The g function uses $\boldsymbol{\kappa} = \{0, 0, 0, 0\}$ and $\mathbf{v} = \{1/1000, 1/1000, 1/1000, 1/1000\}$. Therefore if we use Eq. (19), g_1 adds the magnitude of all the relevant frequencies. If there is a degradation in any of the ball-bearing components, the relevant frequencies will have greater magnitudes and this will be perceived by g_1. Therefore, predicting the hidden state in the testing data can be seen as an approximation of the degradation of the ball-bearing. The idea here is to train models such that they can determine the degradation state of forthcoming ball-bearings. This can be accomplished with the Viterbi paths. In particular for this dataset, we are interested in Bearing 3, since it fails in both the training and testing dataset. Nevertheless, we will train the models for all the bearings and show the scores

(a) Bearing 1   (b) Bearing 2
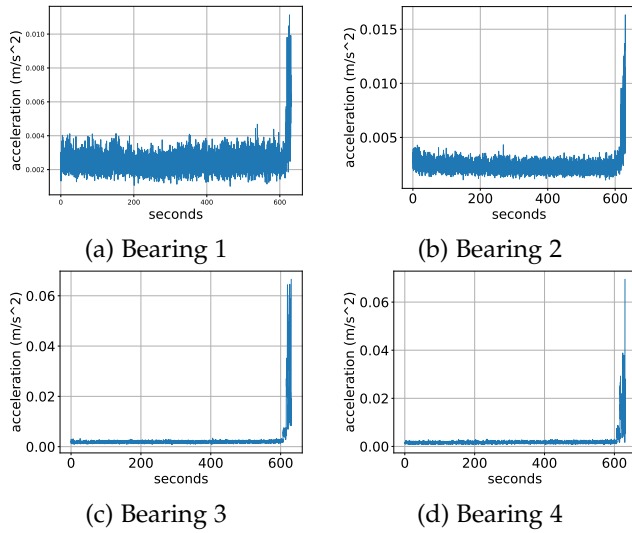
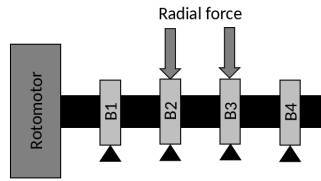(c) Bearing 3   (d) Bearing 4

Fig. 10. Test signals from BPFO

Fig. 11. A rotomotor spins at a speed of 2000 RPM coupled with four Rexnord ZA-2115 ball-bearings. A radial load of 2721.554 kg is applied to B3. A signal record of 0.1 s is taken every twenty minutes. The sampling rate is 20 kHz.

B	Model	LL	BIC	#
B1	AR-AsLG-HMM	**-32095.06**	**64793.79**	57
	AsLG-HMM	-33086.47	66662.88	44
	LMSAR	-43531.13	87727.18	76
	AR-MoG-HMM	-37134.66	75424.18	132
	MoG-HMM	-42949.20	86738.30	96
	Naïve-HMM	-47619.04	95658.03	**36**
	BMM	-41397.04	83633.98	96
	VAR-MVGHMM	-35093.25	72338.73	246
B2	AR-AsLG-HMM	-38443.65	77438.49	51
	AsLG-HMM	-39527.63	79571.46	47
	LMSAR	**-29191.95**	**59048.81**	76
	AR-MoG-HMM	-31924.73	65004.32	132
	MoG-HMM	-37927.80	76695.49	96
	Naïve-HMM	-37296.65	75013.25	**36**
	BMM	-38705.65	78251.19	96
	VAR-MVGHMM	-35674.99	73502.21	246
B3	AR-AsLG-HMM	-56375.55	113424.77	65
	AsLG-HMM	-103975.42	208510.78	52
	LMSAR	**-44120.05**	**88800.04**	64
	AR-MoG-HMM	-44835.74	90616.38	108
	MoG-HMM	-107638.87	216117.64	96
	Naïve-HMM	-119225.39	238870.73	**36**
	BMM	-154597.02	310033.95	96
	VAR-MVGHMM	-108390.68	218513.65	198
B4	AR-AsLG-HMM	-32480.33	65748.06	78
	AsLG-HMM	-43628.14	87833.7	54
	LMSAR	-40498.35	81661.61	76
	AR-MoG-HMM	-38785.03	78724.91	132
	MoG-HMM	-42247.63	85335.15	96
	Naïve-HMM	-49034.67	98489.29	**36**
	BMM	-42300.06	85440.01	96
	VAR-MVGHMM	**-31443.86**	**65039.97**	246

TABLE 8
Model scores for ball-bearing data

obtained in the testing dataset. Additionally we show the Viterbi paths of Bearing 3 to see the respective degradation.

During the training time, the iterations of LMSAR and BMM had to be tuned to prevent problems with the variances or covariance matrices. Additionally, for the BMM, no structural optimization was performed, since it was unfeasible in time.

Table 8 shows the results obtained by the models for each ball-bearing. We observe that the best results in BIC were achieved by different models, say: AR-AsLG-HMM, LMSAR and VAR-MVGHMM. The worst results were attained generally by naïve-HMM and BMM. We observe as well that MoG-HMM and AsLG-HMM got fair results but always worse than their AR counterparts (AR-MoG-HMM and AR-AsLG-HMM, respectively). In particular, in the case of B3, we observe that the use of AR parameters improved significantly the LL and BIC scores. In terms of the number of parameters, we see that naïve-HMM, AsLG-HMM and AR-AsLG-HMM used the least amount of parameters for all the ball-bearings. The remaining models used two or three times the amount of parameters used by naïve-HMM. This implies that AR-AsLG-HMM fulfils its purpose of being a model which uses a reasonable amount of parameters with a good fit for new data.

Fig. 12 shows the paths for the testing B3. We can observe that AR-AsLG-HMM, AsLG-HMM, LMSAR and MoG-HMM exhibit the expected behaviour of the bearings degradation, since they maintain low $g_1(i)$ values during most of the bearing signal and $g_1(i)$ grows abruptly at the end of the bearing life. The models AR-MoG-HMM, Naive-HMM, BMM and VAR-MVGHMM show pure noise or non consistent Viterbi paths, i.e., the $g_1(i)$ function shows high values at the middle of the bearing signal and reduces its values at the end of the bearing life. In the case of LMSAR, the desired behaviour is observed but the differences in the $g_1(i)$ function between the end of the bearing life and the rest of the bearing signal are not significant which affects the model predictive power.

A relevant part of the proposed model is the generation of context-specific Bayesian networks. In Fig. 13 we observe two context-specific Bayesian networks. Fig. 13(a) represents a good health state. In this graph we observe that the cage frequencies (FTF) determine the remaining variables. This implies that knowing the behaviour of the cage, determines the behaviour of the ball-bearing rollers and races. Fig. 13(b) represents a bad health state and shows a more complex structure. In this context-specific Bayesian network AR values are relevant and are taken into consideration. We again see the dominance of the ball-bearings cage (FTF) to determine the dynamical process of the model, but some frequencies are not directly dependent on this variable e.g., the outer race frequencies (BPFO) depend on the inner race frequencies (BPFI) and the roller frequencies (BSF) and these depend directly on the cage frequency (FTF). In summary, these graphs are capable of explaining the ball-bearings dynamical process depending on its health.
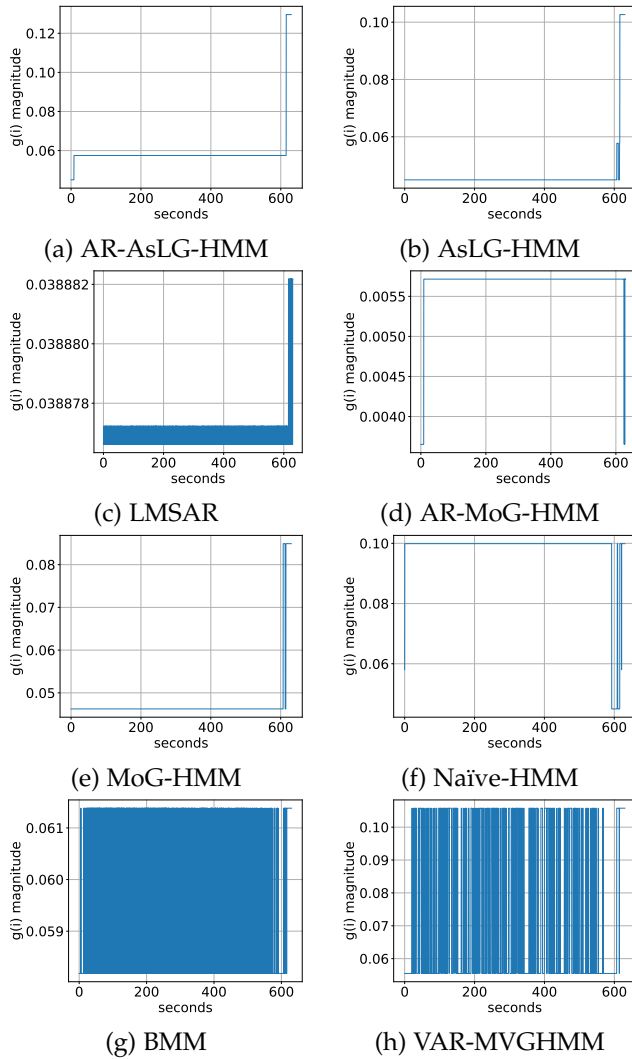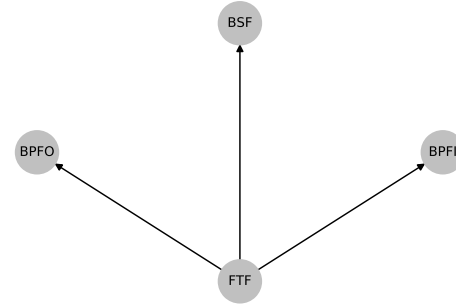
(a) AR-AsLG-HMM   (b) AsLG-HMM

(c) LMSAR   (d) AR-MoG-HMM

(e) MoG-HMM   (f) Naïve-HMM

(g) BMM   (h) VAR-MVGHMM

Fig. 12. Sequence of hidden states by each model for B3

(a) $g(i) = 0.03$

(b) $g(i) = 0.13$

Fig. 13. Context-specific graphs learned by AR-AsLG-HMM. (a) shows a graph where the bearings health is good and (b), where the bearings health is bad.

6 CONCLUSIONS

In this paper, we extended the development of asymmetric hidden Markov models allowing us to determine and learn the optimal number of time lags depending on the value of the hidden state via the SEM algorithm. Also we introduced a greedy-forward heuristic to find the best structure for the model. We also theoretically adapted the forward-backward, Viterbi and EM algorithms to our proposed log-likelihood function. Additionally, we showed that every iteration of the EM algorithm improves the log-likelihood of the model. We introduced a numerical labelling function, which can be helpful in determining the nature of the learned hidden Markov models and to identify changes in the magnitude of the hidden variable.

We used synthetic and real data to validate the proposed model. We compared ourselves with many other models. In general, the AR-AsLG-HMM obtained good results in scores and predictions for synthetic data and real data. We also showed the use of the learned context-specific Bayesian networks to extract information about the nature of the problem being modeled which is harder to obtain from traditional HMMs. Additionally, the number of parameters learned by AR-AsLG-HMM were usually in an intermediate point between the simplest model (naïve-HMM) and the mixture models which as stated before is helpful to prevent data overfitting.

In future work, we would like to combine the idea of asymmetric autoregressive models with other types of HMMs such as HSMMs or HHMMs. Finally, we want to apply the proposed model to online environments and observe its behaviour to detect and treat concept drifts.

REFERENCES

[1] L. R. Rabiner, "A tutorial on hidden Markov Models and selected applications in speech recognition," in *Readings in Speech Recognition*, pp. 267–296, Morgan Kaufmann, 1990.

[2] L. P. Heck and J. H. McClellan, "Mechanical system monitoring using hidden Markov models," in *Proceedings International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1697–1700, 1991.

[3] M. Stanke, O. Schöffmann, B. Morgenstern, and S. Waack, "Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources," *BMC Bioinformatics*, vol. 7, no. 1, p. 62, 2006.

[4] S. Kirshner, S. Padhraic, and R. Andrew, "Conditional Chow-Liu tree structures for modeling discrete-valued vector time series," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 317–324, AUAI Press, 2004.

[5] M. L. Bueno, A. Hommersom, P. J. Lucas, and A. Linard, "Asymmetric hidden Markov models," *International Journal of Approximate Reasoning*, vol. 88, pp. 169–191, 2017.

[6] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller, "Context-specific independence in Bayesian networks," in *Proceedings of the 12th International Conference on Uncertainty in Artificial Intelligence*, pp. 115–123, Morgan Kaufmann., 1996.

[7] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. Wiley, 1976.

[8] J. D. Hamilton, "Analysis of time series subject to changes in regime," *Journal of Econometrics*, vol. 45, no. 1, pp. 39–70, 1990.

[9] A. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, pp. 1291–1294, 1982.

[10] B.-H. Juang and L. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 6, pp. 1404–1413, 1985.

[11] 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.

[12] P. Kenny, M. Lennig, and P. Mermelstein, "A linear predictive HMM for vector-valued observations with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 220–225, 1990.

[13] J. D. Bryan and S. E. Levinson, "Autoregressive hidden Markov model and the speech signal," *Procedia Computer Science*, vol. 61, pp. 328–333, 2015.

[14] E. Nakamura, P. Cuvillier, A. Cont, N. Ono, and S. Sagayama, "Autoregressive hidden semi-Markov model of symbolic music performance for score following," in *16th International Society for Music Information Retrieval Conference* , pp. hal–01183820, Archive ouverte HAL, 2015.

[15] J. Cheng, "A transitional Markov switching autoregressive model," *Communications in Statistics - Theory and Methods*, vol. 45, no. 10, pp. 2785–2800, 2016.

[16] N. Malešević, D. Marković, G. Kanitz, M. Controzzi, C. Ciprianiand, and C. Antfolk, "Vector autoregressive hierarchical hidden Markov models for extracting finger movements using multichannel surface EMG signals," *Complexity*, vol. 2018, pp. 0–12, 2018.

[17] A. Asahara, K. Maruyama, and R. Shibasaki, "A mixed autoregressive hidden Markov chain model applied to people's movements," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 414–417, ACM, 2012.

[18] M. Seifert, K. Abou-El-Ardat, B. Friedrich, B. Klink, and A. Deutsch, "Autoregressive higher-order hidden Markov models: Exploiting local chromosomal dependencies in the analysis of tumor expression profiles," *PLoS One*, vol. 9, no. 6, pp. 1–16, 2014.

[19] I. Stanculescu, C. K. I. Williams, and Y. Freer, "Autoregressive hidden Markov models for the early detection of neonatal sepsis," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1560–1570, 2014.

[20] S. Dang, S. Chaudhury, B. Lall, and P. K. Roy, "Autoregressive hidden Markov model with missing data for modelling functional MR imaging data," in *Proceedings of the 10th Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 93:1–93:8, ACM, 2016.

[21] D. Heckerman, "Probabilistic similarity networks," *Networks*, vol. 20, no. 5, pp. 607–636, 1990.

[22] D. Geiger and D. Heckerman, "Knowledge representation and inference in similarity networks and Bayesian multinets," *Artificial Intelligence*, vol. 82, no. 1, pp. 45–74, 1996.

[23] J. A. Bilmes, "Buried Markov models: a graphical-modeling approach to automatic speech recognition," *Computer Speech and Language*, vol. 17, no. 2, pp. 213–231, 2003.

[24] J. Q. Smith and P. E. Anderson, "Conditional independence and chain event graphs," *Artificial Intelligence*, vol. 172, no. 1, pp. 42–68, 2008.

[25] H. Nyman, J. Pensar, T. Koski, and J. Corander, "Stratified graphical models - Context-specific independence in graphical models," *Bayesian Analysis*, vol. 9, pp. 883–908, 12 2014.

[26] L. M. Barclay, R. A. Collazo, J. Q. Smith, P. A. Thwaites, and A. E. Nicholson, "The dynamic chain event graph," *Electronic Journal of Statistics*, vol. 9, no. 2, pp. 2130–2169, 2015.

[27] C. Puerto-Santana, C. Bielza, and P. Larrañaga, "Asymmetric hidden Markov models with continuous variables," in *Lecture Notes in Artificial Intelligence*, vol. 11160, pp. 98–107, Springer, 2018.

[28] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[29] R. D. Shachter, "Probabilistic inference and influence diagrams," *Operations Research*, vol. 36, no. 4, pp. 589–604, 1988.

[30] K. P. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

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

[33] R. D. Shachter and C. R. Kenley, "Gaussian influence diagrams," *Management Science*, vol. 35, no. 5, pp. 527–550, 1989.

[34] F. Glover and M. Laguna, *Tabu Search*. Springer US, 1997.

[35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[36] M. Martínez and L. Sucar, "Learning dynamic naive Bayesian classifiers.," *Proceedings of the 21th International Florida Artificial Intelligence Research Society Conference*, pp. 655–659, 2008.

[37] S. Zhang, B. Guo, A. Dong, Z. Xu, and S. Chen, "Cautionary tales on air-quality improvement in Beijing," *Proceedings of the Royal Society A*, vol. 473, no. 2205, p. 20170457, 2017.

[38] C. Vitolo, M. Scutari, M. Ghalaieny, A. Tucker, and A. Russell, "Modeling air pollution, climate, and health data using Bayesian networks: A case study of the English regions," *Earth and Space Science*, vol. 5, no. 4, pp. 76–88, 2018.

[39] T. Vairo, M. Lecca, E. Trovatore, A. Reverberi, and B. Fabiano, "A Bayesian belief network for local air quality forecasting," *Chemical Engineering Transactions*, vol. 74, pp. 271–276, 05 2019.

[40] R. Yang, F. Yan, and N. Zhao, "Urban air quality based on Bayesian network," in *2017 IEEE 9th International Conference on Communication Software and Networks*, pp. 1003–1006, 2017.

[41] W. Sun, H. Zhang, A. Palazoglu, A. Singh, W. Zhang, and S. Liu, "Prediction of 24-hour-average $PM_{2.5}$ concentrations using a hidden Markov model with different emission distributions in Northern California," *The Science of the Total Environment*, vol. 443C, pp. 93–103, 2013.

[42] P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. Puerto-Santana, and C. Bielza, *Industrial Applications of Machine Learning*. CRC press, 2018.

[43] Y. Qian, R. Yan, and R. X. Gao, "A multi-time scale approach to remaining useful life prediction in rolling bearing," *Mechanical Systems and Signal Processing*, vol. 83, pp. 549–567, 2017.

[44] Y. Wang and M. Liang, "An adaptive SK technique and its application for fault detection of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 25, pp. 1750–1764, 2010.

Carlos Puerto-Santana Carlos Puerto-Santana received his bachelor's degree in Mathematics from Universidad de los Andes, Bogota, Colombia, in 2016 and his master's degree in Artificial Intelligence from Universidad Politécnica de Madrid, Spain, in 2018. He is currently a doctoral student at Universidad Politécnica de Madrid, Spain and researcher and developer in Aingura IIoT at San Sebastian, Spain.

Pedro Larrañaga Pedro Larrañaga received the M.Sc. degree in Mathematics (Statistics) from the University of Valladolid and the Ph.D. degree in Computer Science from the University of the Basque Country (excellence award). He has been a Full Professor in Computer Science and Artificial Intelligence with the Universidad Politécnica de Madrid (UPM), since 2007. Before moving to UPM, his academic career developed at the University of the Basque Country (UPV-EHU) at several faculty ranks: Assistant Professor, from 1985 to 1998, Associate Professor, from 1998 to 2004, and a Full Professor, from 2004 to 2007. He received the distinction for Full Professor, in 2003. He has published over 200 papers in high-impact factor journals, He has supervised over 30 Ph.D. theses. His research interests include the areas of probabilistic graphical models, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience, industry 4.0, and sports. He is Fellow of the European Association for Artificial Intelligence, since 2012, and a Fellow of the Academia Europaea, since 2018. He has received the 2013 Spanish National Prize in computer science and the prize of the Spanish Association for Artificial Intelligence, in 2018.

Concha Bielza Concha Bielza received her M.S. degree in Mathematics from Universidad Complutense de Madrid, Madrid, Spain, in 1989 and her Ph.D. degree in Computer Science from Universidad Politécnica de Madrid, Madrid, in 1996 (extraordinary doctorate award). She is currently (since 2010) a Full Professor of Statistics and Operations Research with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid.

Her research interests are primarily in the areas of probabilistic graphical models, decision analysis, metaheuristics for optimization, data science, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience, industry and sports analytics. She has published more than 130 papers in impact factor journals and has supervised 17 PhD theses. She was awarded the 2014 UPM Research Prize and the 2020 machine learning award of Amity University (India).