International Journal of Approximate Reasoning 88 (2017) 169–191

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Asymmetric hidden Markov models

Marcos L.P. Bueno [a,*], Arjen Hommersom [a,b], Peter J.F. Lucas [a,c], Alexis Linard [a]

[a] Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands
[b] Faculty of Management, Science and Technology, Open University, The Netherlands
[c] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands

ARTICLE INFO

Article history:
Received 14 October 2016
Received in revised form 24 May 2017
Accepted 26 May 2017
Available online 7 June 2017

Keywords:
Hidden Markov models
Bayesian networks
Model selection
Unsupervised learning
Time series
Structure learning

ABSTRACT

In many problems involving multivariate time series, hidden Markov models (HMMs) are often employed for modeling complex behavior over time. HMMs can, however, require large number of states, what can lead to poor problem insight and model overfitting, especially when limited data is available. In this paper, we further investigate the family of asymmetric hidden Markov models (HMM-As), which generalize the emission distributions to arbitrary Bayesian-network distributions, allowing for state-specific graphical structures in the feature space. As a consequence, HMM-As are able to render more compact state spaces, thus from a learning perspective HMM-As can better handle the complexity-overfitting trade-off. In this paper, we study representation properties of asymmetric and symmetric HMMs, as well as provide a learning algorithm for HMM-As. We provide empirical results based on simulations for comparing HMM-As with symmetric and other asymmetry-aware models, showing that modeling more general asymmetries can be very effective. We also consider real-world datasets from several domains, aiming to show that multiple graphical structures underlying data can be identified and are able to provide additional problem insight. Although learning HMM-As can be more complex, it is shown that it is feasible in practice due to their ability to maintain compact state spaces, yet more expressive ones.

1. Introduction

In many dynamic systems, complex patterns of observations are emitted over time. It is often the case that parts of the underlying process are not observed, e.g. because it is too difficult or impossible. This situation imposes challenges for capturing the interactions between observable features. Hidden Markov models (HMMs) are often employed as models for dynamic systems, having been successful in speech recognition and synthesis domain [1–3]. HMMs have also been applied to problems such as gene prediction and biological sequences [4,5], information retrieval [6,7], and business processes [8]. However, it has been also recognized that HMMs might face limitations to properly capture distributions when limited data is available [9,10,3]. Furthermore, HMMs in practice often have a single chain of states and impose a naive structure over the feature space, which on the one hand alleviates learning and inference costs, but on the other hand gives rise to larger state spaces that can lead to learning issues (e.g. model overfitting) and unsatisfactory problem insight.

Research has been dedicated to extending HMMs for representing more structural information, aiming to render more useful and accurate models, e.g. factorial HMMs [11], input-output HMMs [27], hierarchical HMMs [12], HMM/BN [3], and

* Corresponding author.
E-mail addresses: mbueno@cs.ru.nl (M.L.P. Bueno), arjenh@cs.ru.nl (A. Hommersom), peterl@cs.ru.nl (P.J.F. Lucas), a.linard@cs.ru.nl (A. Linard).

http://dx.doi.org/10.1016/j.ijar.2017.05.011
0888-613X/

autoregressive HMMs [13]. Nevertheless, these extensions cannot capture more specialized (in)dependences, sometimes referred to as asymmetric (in)dependences or local structure [14], i.e. (in)dependences that hold for subsets of values of the involved variables. In the context of graphical models, the representation of asymmetries dates back to Bayesian multinets [14] and similarity networks [15], and had its importance recognized for allowing more efficient probabilistic inference [16–18], better learning [19,20], and for improving problem insight [21] as well.

In the context of HMMs, however, research on capturing asymmetries has been much more limited, with a focus mostly on autoregressive models: modeling higher-order autoregressive interactions (dynamic multinets [22]), tree-based interactions on the observables space (Chow–Liu HMMs [21]), and a combination of first-order autoregressions and tree-based interactions (conditional Chow–Liu HMMs [21]). Therefore, a model able to capture more general asymmetries on the observables space is needed. To address these research aspects, *asymmetric hidden Markov models* (HMM-As) were proposed [23]. We further investigate in this paper the family of HMM-As, in which observations are emitted according to state-specific Bayesian-network distributions, thus representing independences that are not represented in symmetric HMMs.

The contributions of this paper are as follows[1]. We first define HMM-As, then compare their representation aspects with those of symmetric HMMs. We present a learning algorithm for HMM-As, which is based on the structural expectation-maximization framework [24,25], and analyze computational costs associated to symmetric and asymmetric HMMs. A set of varied simulations is then presented, with special attention to the effect of different dataset sizes and number of underlying structured states in learning. Finally, we discuss experiments based on real-world datasets, where we take a close look at the obtained models in order to gain additional insight supported by HMM-As. Such empirical results indicate that HMM-As can be successfully used to obtaining new insight from real-life problems from several domains, including business processes, monitoring of urban pollution, and industrial processes.

2. Preliminaries and hidden Markov models

In this section, we fix the notation and discuss architectures of HMMs and the representation of distribution asymmetries. We denote random variables by upper case letters and sets by bold upper case letters, as in X and $\mathbf{X}$ respectively. The set of values that a random variable X takes on is denoted by $\mathrm{dom}(X)$, where each value from this set is denoted by a lower case letter, as in x (or $\mathbf{x}$, in the case of multidimensional variables). Variables indexed by a time interval $(t_1, \ldots, t_2)$ are indicated as, e.g., $\mathbf{X}^{(t_1:t_2)}$.

2.1. Model architectures

Problems involving time series and sequential processes can be modeled by many probabilistic models, where HMMs are amongst the most used and successful ones [26,2]. In a general problem setting, a set of n observable features $\mathbf{X} = \{X_1, \ldots, X_n\}$ is repeatedly observed over a discrete time horizon $\{0, \ldots, T\}$, and we assume that there is a set of state variables $\mathbf{S} = \{S_1, \ldots, S_r\}$ that we do not observe and are involved in the generation of $\mathbf{X}$ over time. In such problem, we are interested in obtaining a model over $(\mathbf{X}^{(0:T)}, \mathbf{S}^{(0:T)})$ that can be constructed and used feasibly, and yet is realistic and insightful. To this end, different sets of assumptions are very often used, taking also into account domain characteristics. As a consequence, the variety of existing HMMs renders different probabilistic interactions between state variables, observed variables, and observed and states variables (by interaction we refer to (unconditional) probabilistic dependence).

A general HMM is illustrated in Fig. 1, where the exact form of interactions within states and within observables is abstracted. The so-called *independent* HMM (HMM-I) captures the interactions denoted by solid lines, which are typically regarded as the basic interactions covered in any HMM. Furthermore, HMM-Is have a single hidden state per instant (i.e. $\mathbf{S} = \{S\}$) and the observables are conditionally independent given the state, i.e. $P(X_i \mid X_j, S) = P(X_i \mid S)$ whenever $P(X_j, S) > 0$, for all $i, j = 1, \ldots, n$, $i \neq j$. This is usually indicated by $(X_i \perp\!\!\!\perp_P X_j \mid S)$, where $\perp\!\!\!\perp_P$ denotes the independence relation.

From the general HMM architecture previously discussed, other HMMs can be derived, which we summarize in Fig. 2. The emissions of a *standard* HMM (also known in the literature by different terminology, e.g. HMM/BN [3]) can be defined as:

$$P(\mathbf{X} \mid S) = \prod_{i=1}^{n} P(X_i \mid S, \mathrm{Pa}^{-}(X_i)) \tag{1}$$

where $\mathrm{Pa}^{-}(X_i)$ denotes the set of parents of X_i excluding the state, and it depends on the Bayesian network associated to the feature space.

The models from Fig. 2 can be distinguished based on a number of factors, where probabilistic modularity, i.e. the sets of variables that are directly connected in the model, has a major importance. For example, the very high modularity of HMM-Is requires that the state variable summarize more history information, what can imply larger state spaces [11], as opposed to autoregressive HMMs, where the direct interaction between observables is prone to reduce the burden over the hidden states. On the other hand, as autoregressive HMMs tend to have more parameters, they might overfit more

[1] The results and experiments presented in this paper significantly extend an earlier conference paper [23].

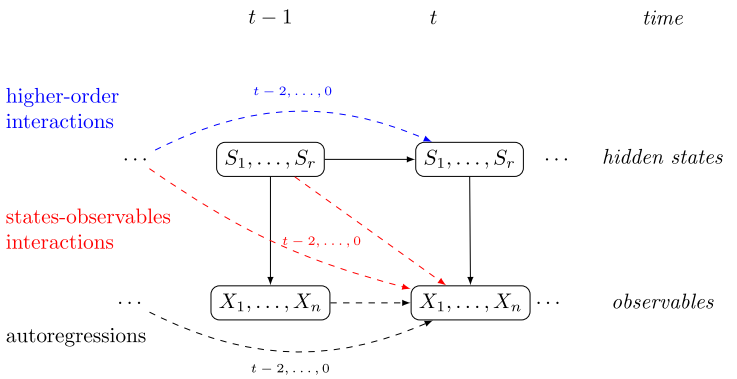


Fig. 1. An abstracted general HMM with hidden states $\{S_1,\ldots,S_r\}$ and observables $\{X_1,\ldots,X_n\}$. Solid arcs indicate interactions present in the independent HMM and related models.

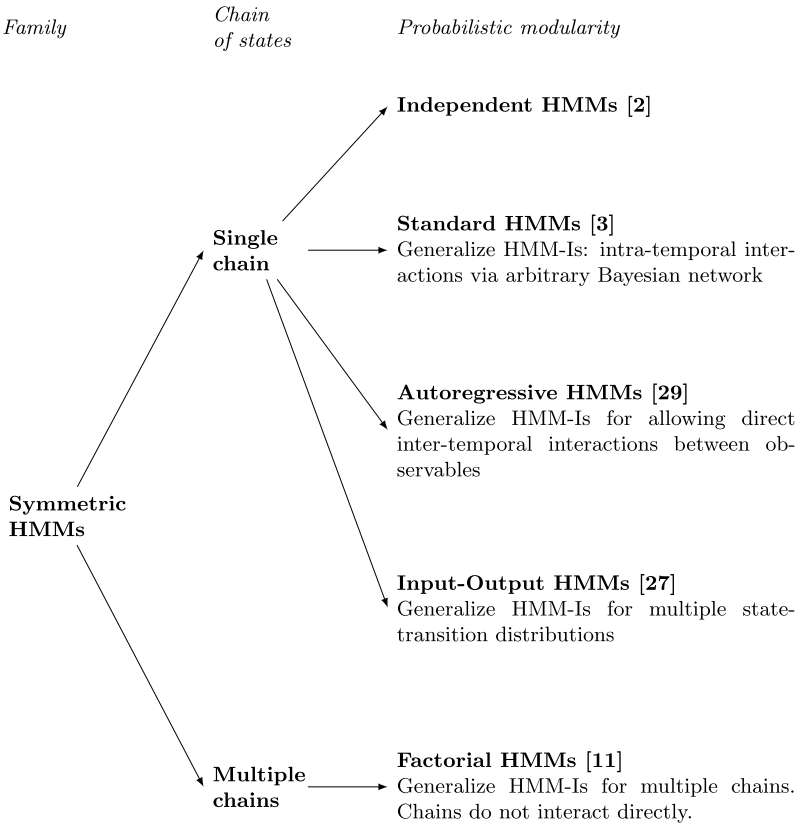


Fig. 2. HMM families which do not represent independence asymmetries. *Intra-temporal* and *inter-temporal* interactions refer to probabilistic interaction between observables from the same time point and between observables from distinct time points respectively.

easily. This illustrates that there can be several trade-offs when a decision must be made about the design of an HMM. The models from Fig. 2 will be denoted here as *symmetric* hidden Markov models, and they usually share the property that the model structure is typically known in advance. As a consequence, learning requires parameter estimation only, which is already intractable in its exact form [2]. Nevertheless, for the case of independent HMMs, good-quality local optima can be obtained using learning methods such as expectation maximization [25], since each of the expectation and maximization iterations can be solved in closed form, which is not the case for more general models, e.g. factorial HMMs [11]. Hence, in practice, the computational cost varies when learning the models of Fig. 2, despite having fixed structure.

Table 1
HMM families which represent independence asymmetries.

Model	Chain of states	Distribution asymmetries	Learning
Dynamic Multinet [22]	Single	Higher-order autoregressions between observables. No intra-temporal correlations.	Discriminative (classification)
Chow–Liu HMM [21]	Single	Intra-temporal interactions modeled by tree distributions.	Generative
Conditional Chow–Liu HMM [21]	Single	First-order autoregressions and tree-based intra-temporal interactions.	Generative
Activator DBN [28]	No chain	Autoregressions and intra-temporal interactions between observables.	Not available
Asymmetric HMM (this paper)	Single	Intra-temporal interactions modeled by arbitrary Bayesian network distributions.	Generative

2.2. Representing asymmetries

The models listed in Fig. 2 deserve further discussion on two points. Little attention has been given to representing more general structure within the feature space of HMMs that can be learned systematically. This is desirable for more domains where a priori knowledge about typical interactions might not be available. Using an inadequate or too restrictive model structure can considerably limit model emissions, which has been considered important for allowing HMMs to properly capture complex distributions [28,21,22]. The second point that we note is that these models do not capture asymmetries in the distribution, i.e. independences that are valid for some values within the variables' domains. Such independences are captured by the notion of context-specific independences, which we define in the following, based on [18].

Definition 2.1. Let P be a probability distribution over the sets of random variables $\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{C}$, which are pairwise disjoint. We say that $\mathbf{V}$ is *contextually independent* of $\mathbf{W}$ given $\mathbf{U}$ and the *context* $\mathbf{c} \in \mathrm{dom}(\mathbf{C})$ if $(\mathbf{V} \perp\!\!\!\perp_P \mathbf{W} \mid \mathbf{c}, \mathbf{U})$ for all values of $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{U}$.

Context-specific independences are able to capture independence statements that are not captured with conditional independence statements. In the context of HMMs, the context is typically given by values of the state variables(s), and we shall refer to such statements as *asymmetric independences* in this paper. We provide a summary of the representation of asymmetries in HMMs in Table 1. For such HMMs, each state $s \in \mathrm{dom}(S)$ determines the parent set of each observable, thus leading to asymmetric independences of the form:

$$(\mathbf{V}^{(t)} \perp\!\!\!\perp_P \mathbf{W}^{(t)} \mid s^{(t)}, \mathbf{U}) \tag{2}$$

where $\mathbf{V}, \mathbf{W} \subseteq \mathbf{X}$. The set $\mathbf{U}$ depends on the state s at t, and it determines the model architecture; for example, in Chow–Liu HMMs $\mathbf{U} \subseteq \mathbf{X}^{(t)}$, whereas in dynamic multinets $\mathbf{U} \subseteq \mathbf{X}^{(0:t-1)}$.

As discussed before, capturing asymmetries is important for inference and learning, however, as Fig. 2 and Table 1 show, research to represent these has been much narrower in the context of HMMs. The sequential nature and the role played by hidden states in HMMs impose other challenges when compared to the static case. Furthermore, while systematic approaches for learning Chow–Liu HMMs are available, their representation of state-specific asymmetries is limited to trees, what can be harmful especially when the feature space has more features, and thus many more structures become available. On the other hand, dynamic multinets directly model potentially longer-history correlations by means of autoregressions, however no instantaneous (i.e. intra-temporal) interactions are modeled, what makes them closer to the original ideas of autoregressive HMMs [29,13] by not fully exploring the graphical structure. In these models, the learning approach is targeted at specific tasks (classification). Thus, there is a need for models that can represent more general asymmetries within the feature space, yet in a compact manner to avoid the need for large amounts of data. Furthermore, the literature lacks a better understanding of the representation capacities of the most used HMM, namely the independent HMM, and other more structured HMMs with respect to state space dimensions and fit quality when the data generation process has structure.

3. Asymmetric hidden Markov models

Asymmetric hidden Markov models generalize HMMs by enriching emission distributions to represent additional qualitative independence per state [23]. In the following we define HMM-As by first defining the association between states and Bayesian-network distributions, followed by a discussion on the representation correspondence between HMM families.

3.1. Model specification

In order to define asymmetries in HMMs, we consider that hidden states induce local models over the observables. This notion can be conveniently represented by conditional Bayesian networks [30], in which a distribution $P(\mathbf{X} \mid S)$ is defined for the observables $\mathbf{X}$ and the state S. As standard conditional BNs provide a single factorization of $\mathbf{X}$ for all $s \in \mathrm{dom}(S)$, we extend this notion for accommodating more general state-specific models as follows.

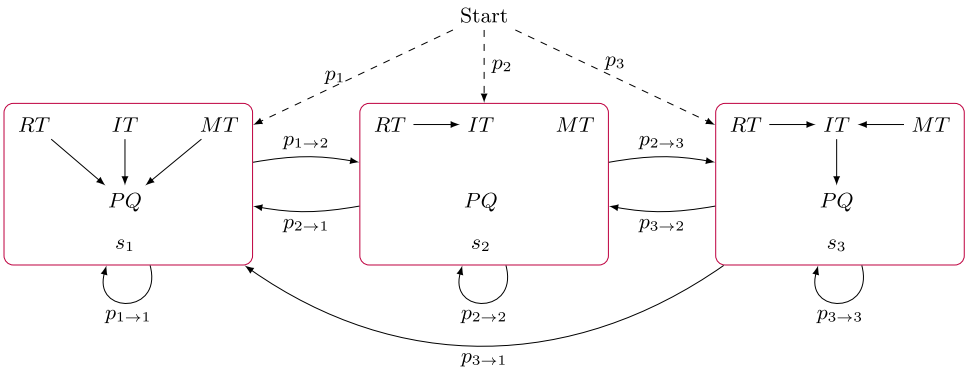


Fig. 3. Probabilistic automaton representation for HMM-A $\mathcal{M}_1$ (dashed arcs indicate initial transitions; zero probabilities are not shown). State-specific BNs are shown in rounded rectangles.

Definition 3.1. Let $\mathbf{X}$ and S be random variables. For each $s \in \mathrm{dom}(S)$, we associate a Bayesian network over $\mathbf{X}$ called *state-specific Bayesian network* for s. If $B_s = (P_s, G_s)$ is the state-specific BN associated to s, we define the following conditional distribution:

$$\begin{aligned} P(\mathbf{X} \mid s) &= P_s(\mathbf{X}) \\ &= \prod_{i=1}^{n} P_s(X_i \mid \mathrm{Pa}_s(X_i)) \end{aligned} \tag{3}$$

where $\mathrm{Pa}_s(X_i)$ denotes the parent set of X_i as dictated by the state-specific BN $B_s = (G_s, P_s)$, in which G_s denotes its graphical structure and P_s its conditional probability tables.

The previous definition maps hidden states to BNs, which conveniently allows for the features in $\mathbf{X}$ to have multiple sets of parents, one set for each state-specific BN.

Definition 3.2. An *asymmetric hidden Markov model* over the random variables $(\mathbf{X}, S)$ is a dynamic system $\lambda = (M_{\to}, M_{\downarrow}, M_0)$, where M_0 is an *initial distribution* $P(S^{(0)})$, $M_{\to}$ is a *transition distribution* $P(S^{(t+1)} \mid S^{(t)})$, and $M_{\downarrow}$ is an *emission distribution* given by

$$P(\mathbf{X}^{(t)} \mid S^{(t)}) = P_{S^{(t)}}(\mathbf{X}^{(t)}) \tag{4}$$

From the definitions shown above, HMM-As are able to explicitly capture more qualitative independences than HMMs. Yet, HMM-As will share a few assumptions with HMMs, namely: the Markovian property and time-invariance. A third assumption that will also hold in HMM-As establishes that the inter-temporal interaction between features must occur via state variables. Hence, given these assumptions, an unrolled HMM-A over the time horizon $\{0, \ldots, T\}$ has the following joint distribution:

$$P(S^{(0:T)}, \mathbf{X}^{(0:T)}) = P(S^{(0)}) \prod_{t=0}^{T-1} P(S^{(t+1)} \mid S^{(t)}) \prod_{t=0}^{T} \prod_{i=1}^{n} P_{S^{(t)}}(X_i^{(t)} \mid \mathrm{Pa}_{S^{(t)}}(X_i)) \tag{5}$$

We note that standard HMMs are therefore special cases of HMM-As, since in the latter every state is associated to the same Bayesian-network structure, i.e. $G_{s_i} = G_{s_j}$ for every $s_i, s_j \in \mathrm{dom}(S)$. An HMM-A can be further visualized as a probabilistic automaton, providing an intuitive representation for states and transitions, as Example 3.1 shows.

Example 3.1. On a regular basis, measurements of print quality (PQ), room temperature (RT), ink type (IT), and media type (MT) are taken from an industrial printer. An HMM-A $\mathcal{M}_1$ for this problem has hidden states that dictate the underlying dynamics, named 'normal', 'failing mode one', and 'failing mode two', denoted by s_1, s_2, and s_3 respectively. $\mathcal{M}_1$ is shown in Fig. 3 as a probabilistic automaton, which runs by alternating taking probabilistic transitions and emitting multivariate observations $(PQ^{(t)}, RT^{(t)}, IT^{(t)}, MT^{(t)})$ according to the states which it traverses.

3.2. Parameterization

The conditional probability table (CPT) of each observable X_i in HMMs has the form $P(X_i \mid S, \mathrm{Pa}^{-}(X_i))$, where Pa^{-} refers to the parents (excluding the state). On the other hand, in HMM-As observables have their parameters associated to state-specific BNs, whose CPTs do not explicitly show the states. Nevertheless, CPTs in the standard sense can easily be obtained from HMM-As, as illustrated next.

Pa	Parameters
s_1, RT, IT, MT	$\theta_{PQ\|s_1,RT,IT,MT}$
s_2	$\theta_{PQ\|s_2}$
s_3, IT	$\theta_{PQ\|s_3,IT}$

(a) Node PQ

Pa	Parameters
s_1	$\theta_{RT\|s_1}$
s_2	$\theta_{RT\|s_2}$
s_3	$\theta_{RT\|s_3}$

(b) Node RT

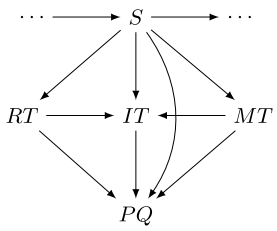
Pa	Parameters
s_1	$\theta_{IT\|s_1}$
s_2, RT	$\theta_{IT\|s_2,RT}$
s_3, RT, MT	$\theta_{IT\|s_3,RT,MT}$

(c) Node IT

Pa	Parameters
s_1	$\theta_{MT\|s_1}$
s_2	$\theta_{MT\|s_2}$
s_3	$\theta_{MT\|s_3}$

(d) Node MT

Fig. 4. Parameterization of probability tables for HMM-A $\mathcal{M}_1$.



(a) Graphical structure (emissions only).

CPT in $\mathcal{M}_1'$	Parameters in $\mathcal{M}_1$
$P(MT \mid S)$	$\theta_{MT\|S}$
$P(RT \mid S)$	$\theta_{RT\|S}$
$P(IT \mid MT, RT, s_1)$	$\theta_{IT\|s_1}$
$P(IT \mid MT, RT, s_2)$	$\theta_{IT\|s_2,RT}$
$P(IT \mid MT, RT, s_3)$	$\theta_{IT\|s_3,MT,RT}$
$P(PQ \mid RT, IT, MT, s_1)$	$\theta_{PQ\|s_1,RT,IT,MT}$
$P(PQ \mid RT, IT, MT, s_2)$	$\theta_{PQ\|s_2}$
$P(PQ \mid RT, IT, MT, s_3)$	$\theta_{PQ\|s_3,IT}$

(b) Conditional probability tables.

Fig. 5. Standard HMM $\mathcal{M}_1'$ that simulates the HMM-A $\mathcal{M}_1$.

Example 3.2. In the HMM-A $\mathcal{M}_1$ (see Example 3.1), the conditional probability tables that are rendered for its feature set are shown in Fig. 4.

Given the HMM-A $\mathcal{M}_1$, it is possible to obtain a standard HMM that represents its distribution over the feature space by turning the asymmetric independences of $\mathcal{M}_1$ into non-asymmetric independences, by taking the minimal directed acyclic graph (DAG) that includes all the dependences of the states in $\mathcal{M}_1$. Let us refer to such a model as *simulating* HMM, which is illustrated next.

Example 3.3. Let $\mathcal{M}_1'$ be a standard HMM for simulating the HMM-A $\mathcal{M}_1$, such that both models have the same number of states. $\mathcal{M}_1$ includes asymmetric independences such as $(PQ \perp\!\!\!\perp RT \mid s_2)$, which does not hold neither in s_1 nor in s_3. In this case, we obtain the conditional dependence $(PQ \not\perp\!\!\!\perp RT \mid S)$, which therefore holds in the simulating HMM $\mathcal{M}_1'$. Similarly, in $\mathcal{M}_1$ it holds that IT and MT are independent in s_1 and s_2 only, hence, it must hold $(IT \not\perp\!\!\!\perp MT \mid S)$ in the simulating HMM. As a consequence, the structure of emissions in $\mathcal{M}_1'$ is denser than that of the state-specific ones from $\mathcal{M}_1$, as it can be noted in Fig. 5.

It is worth noting that now, e.g., the CPT for IT is $P(IT \mid S, RT, MT)$, although direct dependence between IT and $\{RT, MT\}$ exists only when S is s_3 in $\mathcal{M}_1$, which means that redundancies will exist in this CPT, as shown in Fig. 5b. Finally, the total number of independent emission parameters in $\mathcal{M}_1$ is 24 (11 from s_1, 5 from s_2, and 8 from s_3), while in $\mathcal{M}_1'$ it is 42 (obtained by computing the size of the CPT for each variable).

Two points with further implications follow from Example 3.3. As HMM-As allow for savings in the representation size due the direct representation of asymmetries in the distribution, one can readily take advantage of these for speeding up probabilistic inference. Secondly, in HMM-As where a few states induce small amounts of dependences (e.g. state s_2 in $\mathcal{M}_1$), the CPTs of the corresponding standard HMM will be large enough to cover the amount of dependences resulting from the union of all state-specific dependences of the original HMM-A. If there is a great disparity in the amount of asymmetries among the states of the HMM-A, the standard HMM will likely require more probabilistic parameters as well. As a consequence, standard HMMs are prone to reveal much less insight into practical problems.

3.3. Representation aspects

In the following, we discuss how standard and independent HMMs can represent HMM-A distributions, and the effects of such representation on their state spaces and emission distributions.

3.3.1. Relationship with standard HMMs

We provided in Example 3.3 an intuition on how to obtain a standard HMM able to simulate a particular HMM-A, i.e. an HMM that represents the same distribution over the space of observables. Intuitively, the simulating HMM is prone to have

a denser structure compared to the individual states of the original HMM-A, which in the limit reaches a fully connected structure. This is the main idea used in Proposition 3.1 and its proof (see the Appendix), which show that a standard HMM can be obtained from a given HMM-A in the general case. This result also indicates that the simulating HMM does not need additional states for the simulation.

Proposition 3.1. *Let $\mathcal{M}$ be an asymmetric HMM with k states over the observables* $\mathbf{X}$*, where each* $X_i \sim$ Multinomial*,* $i = 1, \ldots, n$*. Then, there exists a* standard *HMM $\mathcal{M}'$ with k states over the same observables which simulates $\mathcal{M}$, i.e.* $P'(\mathbf{X}^{(0:T)}) = P(\mathbf{X}^{(0:T)})$*, where P and P' denote the joint distributions of $\mathcal{M}$ and $\mathcal{M}'$ over* $\mathbf{X}$ *respectively.*

Although the proof of Proposition 3.1 uses an argument based on full connectivity, this is not strictly necessary as the structure on the simulating HMM depends on the amount and form of asymmetries in the original HMM-A. Nevertheless, as Fig. 5b shows, parameter redundancy at the level of states is likely to occur in the standard HMM, preventing inference from readily benefiting from this, as such redundancies are encoded in the CPTs, which is not the case in HMM-As.

3.3.2. Relationship with independent HMMs

While standard HMMs can simulate HMM-As using the same number of states, it is straightforward to see that independent HMMs are not able to do so in the general case. It turns out, however, that the simulation process becomes possible at the cost of expanding the state space of HMM-Is. Intuitively, the more general independence assertions in each state of a given HMM-A must be *broken* into multiple and naively-structured states. We show this result by means of Proposition 3.2.

Proposition 3.2. *Let $\mathcal{M}$ be an asymmetric HMM with k states over the observables* $\mathbf{X}$*, where each* $X_i \sim$ Multinomial*,* $i = 1, \ldots, n$*. Then, there exists an* independent *HMM $\mathcal{M}'$ with k' states over the same observables, such that $\mathcal{M}'$ simulates $\mathcal{M}$ and* $k' \leq k2^n$*.*

The proof of Proposition 3.2 (see Appendix) provides a method for simulating structured distributions of HMM-As by means of HMM-Is, and it also shows an upper bound on the number of states required by the HMM-I. In practice, the amount of dependences per state and the numerical parametrization of the structured model can greatly vary, hence the number of states that a simulating HMM-I requires tends to be lower than the bound, although it can still be much higher than the original number of states of the original HMM-A. Nevertheless, as we further show in this paper, an increase in the state space can be expected when simulating lowly- and moderately-structured distributions.

4. Learning asymmetric hidden Markov models

In this section, we present a learning algorithm for HMM-As. We discuss computational costs for this and other families of HMMs as well.

4.1. Learning setting

In order to learn HMM-As, we assume that state variables are not observed and the graphical structure for emission distributions is unknown. In this case, i.e. learning under missing data and unknown structure, the likelihood function of the observed data is non-decomposable by the graphical structure, which makes analytical methods impossible [30]. The structural expectation-maximization (SEM) [24] is often employed in these settings, which serves as basis for the learning procedure we develop for HMM-As.

The learning setting is score-based and is as follows. Given a dataset $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ of m sequences with duration over $\{0, \ldots, T\}$ complete for a set of features $\mathbf{X}$, and an integer k, we aim to learn an HMM-A with k states that best fits $\mathcal{D}$. The structural EM is an iterative procedure in which each iteration alternates between the expectation and maximization steps (abbreviated as E and M steps respectively), similarly to the standard EM [25]. The E step computes expected sufficient statistics for the states (also known as soft completions) given current model parameters λ and the dataset $\mathcal{D}$, while the M step updates λ using the expected statistics. In the structural EM, an additional step must be considered for updating the model structure as well. In the following, we show how these steps are done for HMM-As, as well as discuss their cost.

4.2. Expectation step

In the E step, we need two expected statistics: the expected occupancy of each state (denoted by γ), and the expected transitions between any two states (denoted by ξ). We further assume that the dataset $\mathcal{D}$ is composed by i.i.d. sequences indexed by $l = 1, \ldots, m$, where each $\mathcal{D}_l$ is an instantiation of $\mathbf{X}^{(0:T)}$, and it will be denoted as $\mathcal{D}^{(0:T,l)}$ when time is needed. Based on the definitions of the expected statistics, it is possible to show that they can be obtained by means of the so-called forward-backward computations as follows (we use a notation based on Rabiner [2]):

$$\begin{aligned}\gamma_t^l(j) &\stackrel{\text{def}}{=} P(S^{(t,l)} = j \mid \mathcal{D}, \lambda) \\ &= P(S^{(t,l)} = j \mid \mathcal{D}_l, \lambda)\end{aligned}$$

$$= \frac{\alpha_t^l(j)\beta_t^l(j)}{\sum_{i=1}^{k}\alpha_t^l(i)\beta_t^l(i)} \tag{6}$$

$$\begin{aligned}\xi_t^l(i,j) &\stackrel{\text{def}}{=} P(S^{(t,l)} = i, S^{(t+1,l)} = j \mid \mathcal{D}, \lambda)\\ &= P(S^{(t,l)} = i, S^{(t+1,l)} = j \mid \mathcal{D}_l, \lambda)\\ &= \frac{\alpha_t^l(i)a_{ij}b_j(\mathcal{D}^{(t+1,l)})\beta_{t+1}^l(j)}{\sum_{p=1}^{k}\sum_{q=1}^{k}\alpha_t^l(p)a_{pq}b_q(\mathcal{D}^{(t+1,l)})\beta_{t+1}^l(q)}\end{aligned} \tag{7}$$

where j indexes state s_j in $\text{dom}(S)$, a_{ij} denotes the transition probability from state s_i to state s_j, and $b_j(\mathcal{D}^{(t+1,l)})$ denotes the emission probability at time $t+1$ for the l-th sequence according to state s_j. We denote by $S^{(t,l)}$ the state for the l-th sequence of $\mathcal{D}$ at instant t.

Equations (6) and (7) suggest that each observation sequence has its own set of expected statistics γ and ξ. Moreover, α and β denote the forward and backward variables respectively, which allow for efficient computation of the expected statistics by means of dynamic programming. The forward and backward variables for each sequence can be computed by:

$$\begin{aligned}\alpha_t^l(j) &\stackrel{\text{def}}{=} P(S^{(t,l)} = j, \mathcal{D}^{(0:t,l)} \mid \lambda)\\ &= \left[\sum_{i=1}^{k}\alpha_{t-1}^l(i)a_{ij}\right] b_j(\mathcal{D}^{(t,l)})\end{aligned} \tag{8}$$

$$\begin{aligned}\beta_t^l(i) &\stackrel{\text{def}}{=} P(\mathcal{D}^{(t+1:T,l)} \mid S^{(t,l)} = i, \lambda)\\ &= \sum_{j=1}^{k} a_{ij}b_j(\mathcal{D}^{(t+1,l)})\beta_{t+1}^l(j)\end{aligned} \tag{9}$$

where we define the basis of recursion as $\alpha_0^l(i) = \pi_i b_i(\mathcal{D}^{(0,l)})$ and $\beta_T^l(i) = 1$ for all $i = 1, \ldots, k$, where π_i denotes the initial probability of s_i. Proposition 4.1 provides the cost of computing the expected statistics needed during an iteration of the E step.

Proposition 4.1. *The computation of one E-step iteration for m sequences in asymmetric HMMs takes $O(mT(kn + k^2))$ time.*

Proof. The dynamic programming procedure for computing the expected statistics of one sequence first stores the emission probabilities of the actual observations, as these values will be repeatedly used during the procedure. We shall store values in a matrix, also known as a lattice (or trellis) structure. First note that the probability of a multivariate observation can be computed in $O(n)$ time, assuming each state-specific BN is conveniently encoded (e.g. using a graph traversal with look-up tables for the parameters). By storing the emission probabilities at each instant by each state in a $(T+1) \times k$ matrix, the total cost for the emissions is $O(Tkn)$.

We proceed to computing the values of the forward and backward variables. The computation of a single α value in Equation (8) amounts to $O(k)$. Hence, the entire lattice for α takes $O(Tk^2)$ time. We build a lattice for β as well, which also takes $O(Tk^2)$ in total.

Once the lattices for α and for β are done, we compute the expected statistic ξ. In Equation (7), observe that its denominator acts as a normalizing factor that depends on t, but not on particular states i or j. Moreover, the computation of a single normalizing factor amounts to $O(k^2)$, hence all such factors take $O(Tk^2)$ time. Back to ξ, we can now compute a single one in constant time, thus the entire trellis for ξ takes $O(Tk^2)$ time. Finally, we can compute the expected statistic γ by means of $\gamma_t(i) = \sum_{j=1}^{k}\xi_t(i,j)$, thus the lattice for γ requires $O(Tk^2)$ time.

The computation of all the lattices for an observation sequence is a sequential process whose total cost is $O(Tkn) + O(Tk^2)$ or $O(T(kn + k^2))$. Hence, the total cost of one E-step iteration for m sequences in HMM-As amounts to $O(mT(kn + k^2))$. □

It is straightforward to see that the cost of the E step in HMM-As is, in fact, the same for several other families of HMMs, including independent and standard HMMs. We also note that the cost is strongly influenced by the number of states (which grows quadratically, whereas the other terms grow linearly).

4.3. Maximization step

In the M step, we update the parameters of the current model $\lambda = (M_{\rightarrow}, M_{\downarrow}, M_0)$ using the expected statistics previously computed. However, as opposed to the standard EM, the M step for HMM-As can no longer be computed efficiently in its

exact form, as the graphical structure on the feature space is unknown, which relates to the intractable problem of finding the optimal structure of a Bayesian network [31]. In fact, this efficiency can only be attained by very few families of HMMs, where the independent HMMs is the main one; even some models that do not capture asymmetries, e.g. the standard HMMs, also lose this property since the structure is unknown as well (although shared by all the states). Thus, one usually relies on approximations to learn reasonable feature–space structures.

In order to devise the update procedure for HMM-As, let us consider the expected score in SEM [32]. The expected score for a candidate model λ^{new} is the expectation of the full-data likelihood taken with respect to the hidden states. As this is based on the current model λ, the expectation is conditioned on λ:

$$\begin{aligned}\mathrm{E}(\lambda^{\text{new}}) &= \mathrm{E}_{\Psi}\left[\log P(\mathbf{X}^{(0:T)}, S^{(0:T)} \mid \lambda^{\text{new}}) \mid \mathbf{X}^{(0:T)}, \lambda\right] - \mathrm{Pen}(\lambda^{\text{new}}) \\ &= \sum_{\Psi} P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) \log P(\mathbf{X}^{(0:T)}, S^{(0:T)} \mid \lambda^{\text{new}}) - \mathrm{Pen}(\lambda^{\text{new}})\end{aligned}$$

where the subscript $\Psi = \mathrm{dom}(S^{(0:T)})$ denotes the domain of the variable(s) over which the expectation is taken. Note that $P(\mathbf{X}^{(0:T)}, S^{(0:T)} \mid \lambda^{\text{new}})$ factorizes according to the structure of the HMM-A (see Equation (5)), thus:

$$\begin{aligned}\mathrm{E}(\lambda^{\text{new}}) &= \sum_{\Psi} P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) \log \left[P(S^{(0)}) \prod_{t=0}^{T-1} P(S^{(t+1)} \mid S^{(t)}) \prod_{t=0}^{T} P(\mathbf{X}^{(t)} \mid S^{(t)}) \right] - \mathrm{Pen}(\lambda^{\text{new}}) \\ &= \sum_{\Psi} \log P(S^{(0)}) P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) + \sum_{\Psi} \left(\sum_{t=0}^{T-1} \log P(S^{(t+1)} \mid S^{(t)}) \right) P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) \\ &\quad + \sum_{\Psi} \left(\sum_{t=0}^{T} \log P(\mathbf{X}^{(t)} \mid S^{(t)}) \right) P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) - \mathrm{Pen}(\lambda^{\text{new}}) \end{aligned} \tag{10}$$

Equation (10) suggests that each term of the expected score can be optimized separately. The result is the parameter updating in the SEM process as follows.

4.3.1. Structure learning

In Equation (10), we identify the term associated to the emissions as:

$$\begin{aligned}&\sum_{\Psi} \left(\sum_{t=0}^{T} \log P(\mathbf{X}^{(t)} \mid S^{(t)}) \right) P(S^{(0:T)} \mid \mathbf{X}^{(0:T)}, \lambda) - \mathrm{Pen}(M_{\downarrow}^{\text{new}}) \\ &= \sum_{j=1}^{k} \sum_{t=0}^{T} \log P(\mathbf{X}^{(t)} \mid S^{(t)} = j) P(S^{(t)} = j \mid \mathbf{X}^{(0:T)}, \lambda) - \mathrm{Pen}(M_{\downarrow}^{\text{new}}) \\ &= \sum_{j=1}^{k} \sum_{t=0}^{T} \gamma_t(j) \log P(\mathbf{X}^{(t)} \mid S^{(t)} = j) - \mathrm{Pen}(M_{\downarrow}^{\text{new}}) \end{aligned} \tag{11}$$

The emissions term (Equation (11)) can be further decomposed *per state*, because the state-specific networks are independent of each other. The advantage now is that *each state-specific network can be locally learned*. In this work, the penalty term is defined according to the BIC score, which penalizes each candidate for both model complexity and for the dataset size [33]. Thus, for a state s_j, its portion of the emissions term is:

$$\sum_{t=0}^{T} \gamma_t(j) \log P(\mathbf{X}^{(t)} \mid S^{(t)} = j) - \mathrm{Pen}(M_{\downarrow}^{\text{new}}; s_j) = \sum_{t=0}^{T} \gamma_t(j) \log P(\mathbf{X}^{(t)} \mid S^{(t)} = j) - \frac{K_j \log(T+1)}{2} \tag{12}$$

which in the case of m sequences becomes:

$$\sum_{l=1}^{m} \sum_{t=0}^{T} \gamma_t^l(j) \log P(\mathcal{D}^{(t,l)} \mid S^{(t,l)} = j) - \frac{K_j \log m(T+1)}{2} \tag{13}$$

where K_j is the number of parameters in the model for state s_j. In HMM-As we wish to learn state-specific graphical structures in the M step, hence we run structure learning for each of the k states separately. In practice, structure learning often relies on approximate methods for exploring the search space of structures in feasible time and yet providing reasonable solutions. These procedures include, e.g., heuristics, bio-inspired methods, integer programming, among others [34–36].

In this work, we considered tabu search [37] (TS) to explore the candidate space of structures, which is a polynomial-time procedure based on hill-climbing search. A TS iteration explores the neighborhood of the current solution (initially an empty network) by means of adding, deleting or reversing an arc from it. The current solution is added to the tabu list (which

stores the 10 most recently explored networks, in this implementation); furthermore, only neighborhood solutions that are not in the tabu list are added to the neighborhood set (initially empty). Once the neighborhood set has been updated, the best of its solutions is taken out and set in the next iteration as the current solution, which is not required to be better than the previous one (however, this is allowed for no more than 10 consecutive iterations). During the search, the respective part of the expected score (Equation (13)) is used to compare candidate structures. Once the stopping criterion is reached (e.g. if the neighborhood set is empty or if more than 10 iterations without improvement have passed), the best structure ever seen is returned. Given the described steps for TS, the cost of each structure learning run is bounded by a polynomial cost on the method's hyperparameters aforementioned and the number of observable features.

4.3.2. Parameter update

After obtaining a model structure for λ^{new}, it is possible to show that maximizing the expected score (Equation (10)) leads to the following update formula for the transition probabilities:

$$\hat{a}_{ij} = \frac{\sum_{l=1}^{m} \sum_{t=0}^{T-1} \xi_t^l(i,j)}{\sum_{l=1}^{m} \sum_{t=0}^{T-1} \gamma_t^l(i)} \tag{14}$$

The update of the emission probabilities, in turn, is more complicated than in standard EM, as the feature space is multivariate and each feature can have other parents beyond the state variable. Furthermore, the parent set for a given feature can vary among states. Nevertheless, we can take advantage of the fact that the state-specific BNs allow us to factorize the joint distribution of the feature set $\mathbf{X}$, thus we can update the probability tables for one variable at a time. For state s_j and feature X, we update its probability tables as follows:

$$\hat{b}_j(X = x, \text{Pa}_j(X) = y) = \frac{\sum_{l=1}^{m} \sum_{t=0}^{T} \gamma_t^l(j) \mathbb{1}(X^{(t,l)} = x, \text{Pa}_j(X)^{(t,l)} = y)}{\sum_{l=1}^{m} \sum_{t=0}^{T} \gamma_t^l(j) \mathbb{1}(\text{Pa}_j(X)^{(t,l)} = y)} \tag{15}$$

where $\mathbb{1}$ is the indicator function. As in the case of arbitrary Bayesian networks, the cost of this calculation strongly depends on the connectivity of the network, being exponential in the number of features in the worst case. However, if the parent sets have moderate sizes, this can be very reasonable in practice.

As a final remark in learning HMM families, we note that a simpler version of this M step is needed for learning standard HMMs. In that case, structure learning is executed only once, as all the states will share a single structure. Analogously, updating the parameters in standard HMMs can still be costly due to the reasons previously discussed.

5. Assessment via simulations

In this section, we aim to understand how unstructured, structured, and asymmetry-aware models cope with data generated from asymmetry-aware distributions. We also intend to analyze the effect of different amounts of data in model quality. To this end, we consider data sampled from HMM-A distributions to simulate different scenarios.

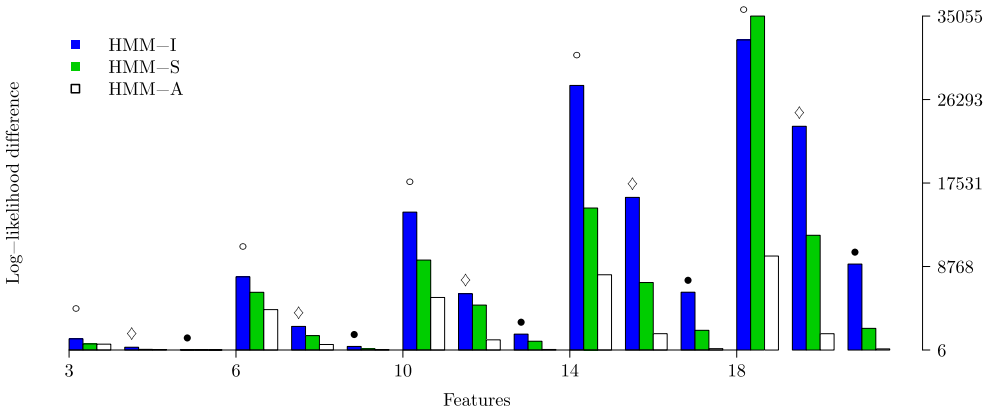
5.1. Model selection

In order to learn models that generalize best, we considered a model selection procedure to determine state spaces balancing complexity and overfitting avoidance as follows. Given a sequence dataset $\mathcal{D}$, models are learned incrementally by increasing the number of states until overfitting is observed, which corresponds to the point where model score no longer increases. Model scoring is based on 10-fold cross-validation: for each fold, a model is learned using training data (90% of the data) and its log-likelihood over validation data (the remaining portion) is computed; after processing all the folds, the mean log-likelihood is taken, corresponding to the final score. To better assess learning, we learn 30 initial models for each k states, and select the one that generalizes the best to represent models with k states. Once the number of states has been determined, the final model L is learned using the entire dataset and those initial parameters, and is evaluated by means of 60 independent datasets (not used in learning nor validation; each independent dataset has 2,000 sequences with length 20 each).

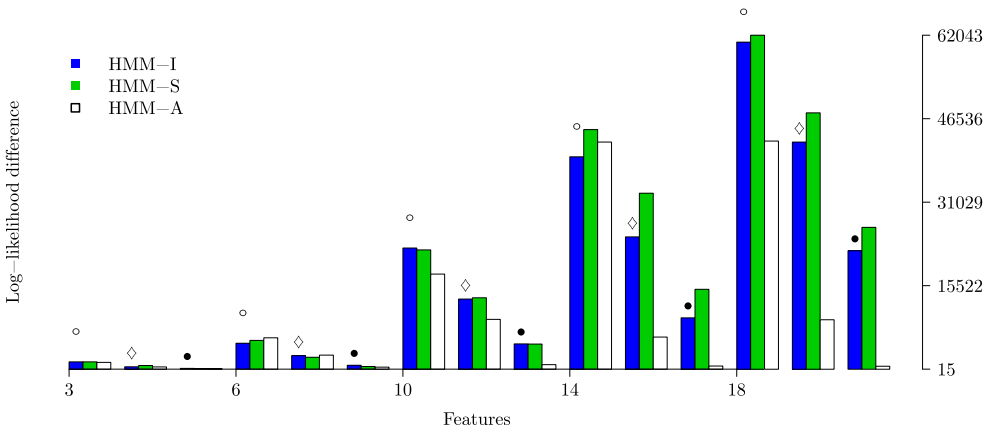
Each learned model L is evaluated by comparing likelihoods as follows. Let R be the true model used to sample $\mathcal{D}$, then we define the fit quality of L as $\text{LL}_R - \text{LL}_L$, where LL_R and LL_L denote the mean log-likelihood of the models R and L over testing data respectively. This fit quality can be interpreted as the logarithm of the number of times the likelihood of the true model R is in comparison with the likelihood of the learned model L (in non-logarithmic scale). Hence, if the difference equals zero, it indicates that L and R fits equally well, while a difference larger than zero indicates that L fits worse than R. Thus, learned models with log-likelihood difference closer to 0 are preferred. We finally note that this procedure allows us to compare models learned with different amounts of data, as they are evaluated over the same testing datasets.

5.2. Datasets

Datasets were sampled from random HMM-As, which were generated taking into account that many real-life networks have an average degree between 2 and 4 per node (i.e. the sum of in- and out-degrees). This is the case, for example, in



(a) Number of states in true models = 2.



(b) Number of states in true models = 6.

Fig. 6. Fit of asymmetric and symmetric HMMs learned in simulations. Datasets sampled from true models have 50 sequences (length 10 time points, ○), 200 sequences (length 10 time points, ◇), and 1,000 sequences (length 20 time points, ●). Note that scales on Y axes differ.

well-known BNs, such as alarm, pathfinder, asia, and insurance [38]. Hence, in order to generate ground truth models having state-specific BNs with a reasonable, and yet realistic structure, the maximum degree of each node on each network was set to 3.

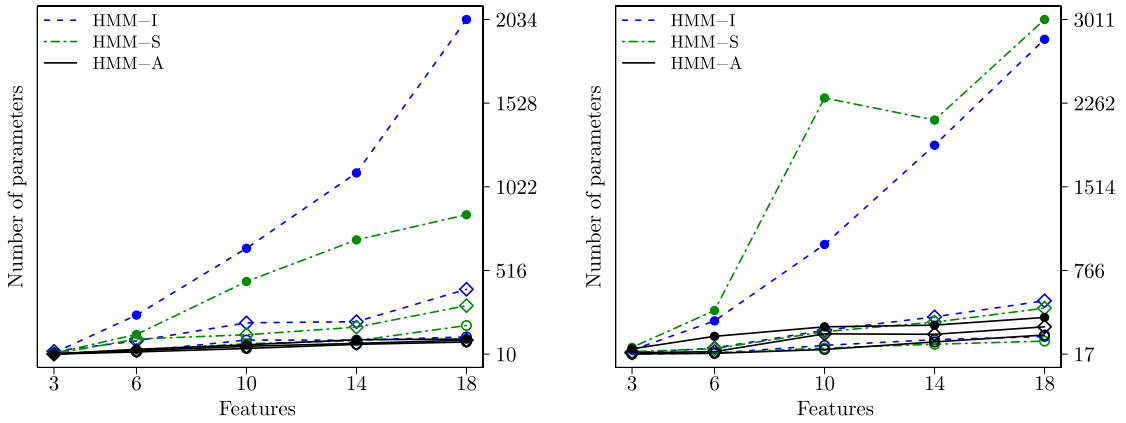
In order to build a random HMM-A with k states, its initial and transition matrices are sampled from Dirichlet distributions with concentration parameters all set to 1. Thus, valid distributions are obtained, i.e. matrices with rows that sum to 1 [39]. The emissions are Bayesian networks made of uniformly sampled DAGs [38,40], whose nodes have the aforementioned maximum degree. All observables are modeled as random variables following Bernoulli distributions, whose parameters are sampled from Dirichlet distributions as before. We note that this procedure is also used to generate the initial models used in learning (see Section 5.1), except that no maximum degree is set. Finally, the constructed scenarios considered the following quantities: number of observables $n \in \{3, 6, 10, 14, 18\}$, state space dimension of true models $k \in \{2, 6\}$, and the number of sequences as 50 (each one with length of 10 time points), 200 (length 10) and 1,000 (length 20).

5.3. Results for symmetric models

Fig. 6 shows the log-likelihood differences between asymmetric and symmetric HMMs based on simulated data (here, HMM-S refers to standard HMM). We first note that, as expected, all the classes of models obtained better fit when more data is provided, what is influenced by the fact that more states can be learned prior to overfitting. The results also suggest that independent and standard HMMs were not able to provide the same model quality as HMM-As, even when the highest amounts of data were provided to all the three models. Hence, it seems reasonable to expect that much more data would be needed in order to learn models that fit as well as the learned HMM-As (in this case, using 1,000 sequences). Concerning the scarcer datasets (note that the larger datasets are 40× larger than the smaller ones), HMM-As achieved superior model quality on most cases. This allows us to conclude that HMM-As showed a good compromise in a varied range of dataset sizes.

Table 2
State spaces of asymmetric and symmetric HMMs learned in simulations.

n	HMM-I	HMM-S	HMM-A	n	HMM-I	HMM-S	HMM-A
(a) Dataset size = 50 sequences.							
3	3	2	2	3	3	3	3
6	3	2	2	6	3	2	2
10	6	2	2	10	6	2	3
14	5	2	2	14	7	2	4
18	5	2	2	18	7	2	5
Number of states in true models = 2				Number of states in true models = 6			
(b) Dataset size = 200 sequences.							
3	4	2	2	3	4	4	4
6	7	3	2	6	6	3	3
10	10	2	2	10	11	3	8
14	9	2	2	14	13	2	6
18	13	2	2	18	15	3	6
Number of states in true models = 2				Number of states in true models = 6			
(c) Dataset size = 1,000 sequences.							
3	3	2	2	3	6	6	6
6	13	2	2	6	15	6	7
10	21	2	2	10	27	6	7
14	27	2	2	14	37	3	6
18	37	2	2	18	45	3	6
Number of states in true models = 2				Number of states in true models = 6			



(a) Number of states in true models = 2. (b) Number of states in true models = 6.

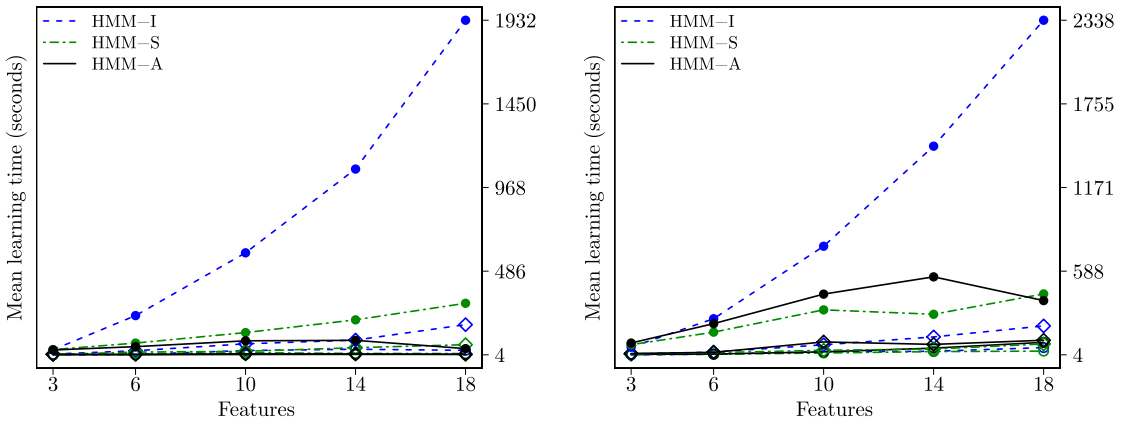
Fig. 7. HMM-As and symmetric HMMs learned from simulated data: number of parameters for different cases. Note that scales on Y axes differ.

In terms of scaling, e.g. when modeling more observables, the additional structure of HMM-As avoided pitfalls that can hinder independent and standard HMMs: HMM-Is will tend to increase their state space, while standard HMMs will tend to model denser feature–space graphical structures. As a consequence, in most cases these symmetric models approach overfitting with much less model quality than HMM-As. In other words, despite the representational equivalence between HMM-As and independent and standard HMMs in theory, such symmetric models can be considered limited in practice. These claims are further supported by results in Table 2 showing the corresponding state space dimensions, and Figs. 7a–7b showing the number of parameters.

Table 2 shows the dimension of state spaces associated to learned models, suggesting that approximating HMM-A distributions required independent HMM with state spaces substantially larger than the true models' spaces, while this was not the case for standard HMMs. Nevertheless, as Figs. 7a–7b show, the number of parameters in these two families were substantially higher than those of learned HMM-As, specially when more features were involved. With regard to running time in learning, Figs. 8a and 8b show that, somewhat surprisingly, learning HMM-Is was more costly in most cases than HMM-As: although learning HMM-As is done via structural EM, its combination with search heuristics and smaller space state was in practice more efficient than the EM used to learn HMM-Is.

5.4. Results for asymmetric models

Fig. 9 shows the fit quality results for HMM-As and Chow–Liu HMMs (HMM-CLs). These results indicate that restricting the feature space to trees prevented HMM-CLs from achieving model quality as high as that of HMM-As on most of the



(a) Number of states in true models = 2. (b) Number of states in true models = 6.

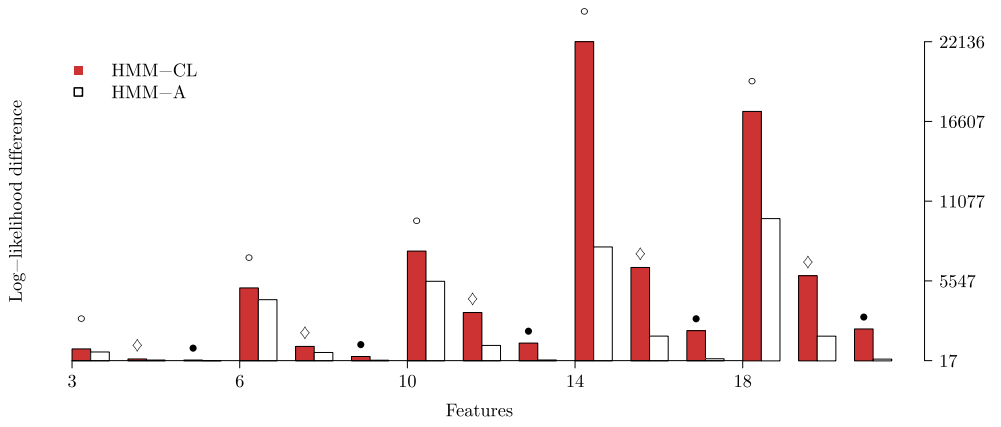
Fig. 8. HMM-As and symmetric HMMs learned from simulated data: mean learning time in seconds.

Table 3
State spaces of HMM-As and HMM-CLs learned in simulations (k denotes number of states).

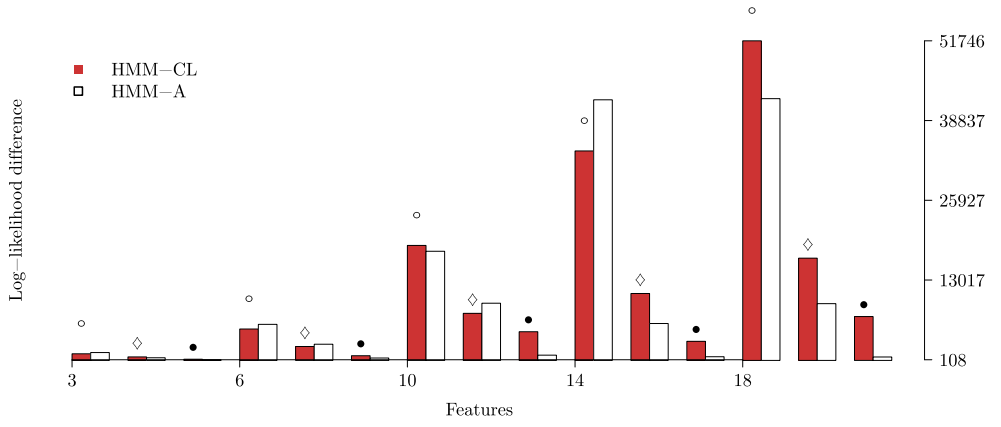
n	HMM-CL	HMM-A	n	HMM-CL	HMM-A
(a) Dataset size = 50 sequences.					
3	2	2	3	3	3
6	2	2	6	2	2
10	2	2	10	3	3
14	3	2	14	4	4
18	2	2	18	4	5
k in true models = 2			k in true models = 6		
(b) Dataset size = 200 sequences.					
3	2	2	3	4	4
6	3	2	6	3	3
10	4	2	10	6	8
14	3	2	14	6	6
18	3	2	18	7	6
k in true models = 2			k in true models = 6		
(c) Dataset size = 1,000 sequences.					
3	2	2	3	7	6
6	4	2	6	8	7
10	7	2	10	7	7
14	9	2	14	15	6
18	8	2	18	21	6
k in true models = 2			k in true models = 6		

considered scenarios. This is more evident in the cases involving more observables, where the learned HMM-As reached the most superior model quality compared to HMM-CLs, what is likely influenced by the size of the possible graphical structures for emissions, situations which HMM-As can better handle, since HMM-As are not restricted to trees. On the other hand, HMM-CLs are prone to be more efficient in practice, since learning Chow–Liu trees can be done efficiently per EM iteration [21]. Similarly to the symmetric models case, HMM-As could be trained with less data, and yet provided similar or better model quality than HMM-CLs – although here to a lesser extent when the data generating process had a higher number of hidden states. Furthermore, extending the state spaces of HMM-CLs resulted in better models, however, prior to overfitting these models achieved lower quality than HMM-As.

A comparison based on Figs. 6 and 9 suggests that modeling state-specific structures, whether by means of general asymmetries as HMM-As or by tree-shaped ones as HMM-CLs, led to better results than those of symmetric models. HMM-As needed in general fewer states or fewer parameters than symmetric HMMs, which also holds for HMM-CLs with respect to symmetric HMMs, as shown in Table 3 and Fig. 10. Hence, the results of this section suggest a somewhat consistent conclusion: capturing the distribution underlying data generated by more structured processes is more adequate by means of models that capture distribution specificities associated to the hidden states. Although symmetric models can in theory capture such distributions, whether by an increase of their state spaces or by modeling denser emissions structure, in many realistic situations – where data is often limited – the asymmetric models exhibited several advantages and better handled the complexity *versus* quality trade-off.

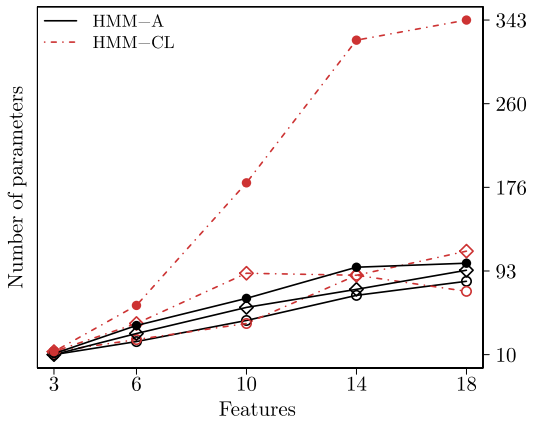


(a) Number of states in true models = 2.

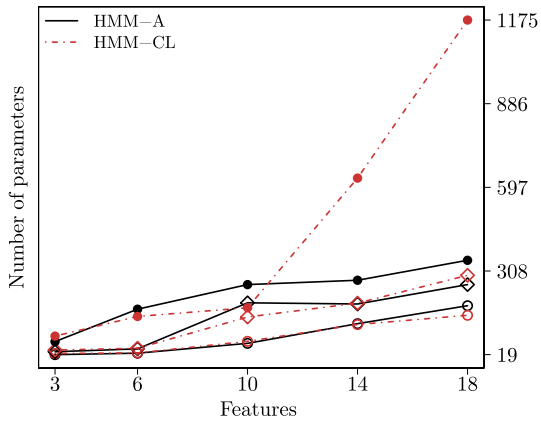


(b) Number of states in true models = 6.

Fig. 9. Fit of asymmetric and Chow–Liu HMMs learned in simulations. Datasets sampled from true models have 50 sequences (length 10 time points, ○), 200 sequences (length 10 time points, ◇), and 1,000 sequences (length 20 time points, ●).



(a) Number of states in true models = 2.



(b) Number of states in true models = 6.

Fig. 10. HMM-As and HMM-CLs learned from simulated data: number of parameters for different cases.

Table 4
Summary of real-world datasets. The *sequence data* column shows the number of sequences together with sequence duration in parenthesis.

Dataset	n	Description	Sequence data
Volvo	3	Event logs of software incidents	151 (50)
Rabobank	6	Event logs of software incidents (interactions subset)	500 (30)
Airquality	12	Urban pollution monitoring	40 (48)
Printer $\mathcal{R}_1$	7	Performance of printing nozzles and maintenance activity	27 (15)
Printer $\mathcal{R}_2$	7	Performance of printing nozzles and maintenance activity	52 (15)
Printer $\mathcal{R}_3$	7	Performance of printing nozzles and maintenance activity	58 (15)

6. Experiments with real-world datasets

In this section, we describe experiments for learning symmetric HMMs, HMM-CLs and HMM-As from real-world datasets originated from several domains. In order to empirically determine state spaces and assess learned models, we used a procedure similar to the one described in Section 5. It differs in that real-life datasets are split in two parts: one for selecting models via 10-fold cross-validation (using 80% of the data), and the remaining portion for assessment of generalization.

6.1. Datasets

The datasets considered in this section are summarized in Table 4 and described next.

6.1.1. Business process data

Business process datasets consist of event-log records on software incidents related to, e.g., software bugs, hardware problems, among others within the scope of ICT company departments. These datasets are often used for process mining, covering tasks such as conformance checking (i.e. checking whether the business process specification complies with the running process), process discovery and process enhancement. Learning business models as done in process mining field often intends to capture the underlying sequential behavior of actions within events. Thus, given a collection of events, business models fit to them in order to represent different ways in which an event can develop over its lifetime [41].

As opposed to business models (e.g. workflow-like models), where one often wants to understand the internals of events, in this section we learn a complementary behavior from event-logs data in the form of influences among events. As this is less evident from data and involves multivariate observations (since events are typically composed of many features), this is a challenging problem, for which this section offers an HMM-based solution. We considered two datasets from the BPI (business process intelligence) challenge, described as follows.

Volvo dataset The Volvo IT Belgium dataset [42] consists of event logs of software incident registered during the period of 2011–2012. Each data point describes an incident by means of three features: incident impact, push to front (i.e. whether the incident was handled by a service desk team or required other specialized teams as well), and country (referring to whether the incident involved employees from different nationalities). The Volvo data was split in sequences such that each sequence has approximately 5 days of incidents.

Rabobank dataset The Rabobank Group ICT dataset [43] consists of event-log records of software incidents over the period of 2011–2014, however from a different software domain than the Volvo dataset. We considered the part of the data related to interactions, which registers the first contact between a user of a software component and a service desk team. An interaction call can lead to an incident or not. Each interaction is described in the Rabobank dataset by a set of six features: type of involved item (e.g. application, hardware, network-related issues), impact (in case of service disruption), priority, category (i.e. whether the event refers to a request for information or an incident), first call (i.e. whether the interaction could be solved by service desk team or led to an incident for further resolution), and handle time (i.e. the amount of time to resolve the service disruption).

Learning HMM-As for business processes aims in first place to provide well fitted models, but also to discover different dynamics that governs the generation of incidents and interactions. This can then be turned into practical knowledge, e.g., to aid decision makers to render more effective and resource-saving business processes. We shall discuss more on this in Section 6.3.

6.1.2. Airquality data

The Airquality dataset contains data on gas pollutants in the context of urban pollution monitoring [44]. The feature set is composed by two different sources of information: a set of reference pollutant concentrations provided by conventional stations, and a set of measurements provided by a multi-sensor device. Originally, the Airquality dataset was used to evaluate and calibrate sensor devices for estimating the concentration of pollutants as technological means with low cost for convenient air monitoring across urban spaces. We considered a feature set with 12 variables corresponding to the original measurements, which were discretized for the experiments in this section.

Table 5
Generalization assessment of learned models on real-world datasets. Notation: k denotes the number of states, NLL the normalized log-likelihood, and #Pa. the number of parameters. Results that generalized the best are bold-faced and followed by an asterisk.

Dataset	HMM-I			HMM-S			HMM-CL			HMM-A		
	k	NLL	#Pa.	k	NLL	#Pa.	k	NLL	#Pa.	k	NLL	#Pa.
Volvo	8	65.2	87	5	**64.7***	59	6	65.2	65	5	**64.7***	50
Rabobank	10	**48.5***	159	5	48.7	214	6	**48.5***	101	4	**48.5***	73
Airquality	8	32.9	159	8	32.1	623	8	35.4	247	6	**31.3***	170
Printer $\mathcal{R}_1$	5	49.8	59	3	48.7	53	3	46.6	47	3	**46.0***	36
Printer $\mathcal{R}_2$	4	60.5	43	3	61.7	65	3	60.4	47	3	**59.9***	43
Printer $\mathcal{R}_3$	4	48.0	43	3	46.0	56	3	46.4	47	3	**45.7***	36

6.1.3. Printers data

We also considered data to support understanding the behavior of modern, complex engineered artifacts (also called cyber-physical systems), for which we use a large-scale printer as a case study. Whereas engineers understand the functioning of the individual components in considerable depth and detail, as a consequence of their intricate design, they find it much more difficult to understand the behavior of the artifacts at a certain level of abstraction, as well as their interaction. In order to learn the temporal behavior of such systems, data was gathered from three printers of the same printer family, where the usage of the printers differs as function of time, and as function of the print jobs being rendered. In this case study, we focus in particular on one component – the nozzle – that aims at jetting ink on the paper [23]. The behavior of nozzles as function of time depends on several factors, like quantity of ink used, time since last maintenance and some environmental parameters.

The logs that were considered consist of a 1-year record of nozzle-related factors continuously monitored. We considered a key maintenance action that is performed by the machine from time to time, and gathered data on nozzle-related components between each maintenance occurrence, such that each (multivariate) observation includes the following features: interval duration (i.e. the length of time since the previous maintenance action), total workload, frequency of another related maintenance action, and color-related features. The goal of our experiment is to discover relations between features and how it influences the proper functioning of the nozzles.

6.2. Results

We first report results on fit quality based on model selection, where Fig. 11 shows the mean validation log-likelihoods in function of the number of states. These results suggests that the structural simplicity of independent HMMs could be compensated to some extent by learning larger state spaces. In these cases, model quality similar to that attained by more structured models (i.e. standard and HMM-As) could be achieved by HMM-Is. However, this was not possible in all the cases, in particular in the business process datasets. In these cases, prior to overfitting the structured models had a much better fit, suggesting that in some cases the presence of non-trivial structure over observables seems crucial to attaining better fit.

With respect to the structured models, contrasting standard HMMs with HMM-As indicates that HMM-As achieved superior fit on some cases and similar model quality on the remaining ones. The learned HMM-As better fitted the data than Chow–Liu HMMs in general as well. It is interesting to note that HMM-CLs impose tree structures to its emissions, something that might not be always beneficial. For example, in the Volvo and Airquality cases, the results suggest that even a symmetric model (HMM-S) was able to provide better results than HMM-CLs, what is interesting as an HMM-S does not necessarily learn a connected structure for the emissions space. In general, it can also be observed from Fig. 11 that structured models as standard HMMs and Chow–Liu HMMs overfit much more easily than HMM-As, suggesting that HMM-As provide more parsimonious solutions to these real problems.

Dynamic Bayesian networks (DBNs) were also learned from the real-world datasets, as shown in Fig. 11. The results indicate that DBNs provided consistently inferior model quality than HMMs. Although these results are not directly related to comparing HMMs, they suggest that modeling autoregressions alone (as in DBNs) is not a guarantee for good fit in real-life datasets: modeling multiple (and possibly structured) distributions via hidden states can be much more powerful, yet no autoregressions are modeled by these HMMs. A question that remains is whether including autoregressions in HMM-As can bring real benefits, which despite being a tempting idea is likely to make such models approach overfitting more easily.

Having discussed the dynamics of model quality based on validation log-likelihoods experiments, we now use these results to select and learn models in order to discuss problem insight, as well as to assess their generalization. To this end, we select models in a flexible way: we pursue models with the highest score, except when there are multiple models with similar quality, in which case we select those with the lowest dimension. After finishing this, we learn models with the selected dimension using the entire datasets and calculate their likelihood over testing data (i.e. data that was not used in cross-validation).

The models learned for generalization assessment are summarized in Table 5. As there is no ground-truth model for the real-world datasets, to facilitate comparison we used normalized log-likelihoods $\text{NLL} \stackrel{\text{def}}{=} -\text{LL}/c$, where LL is the log-likelihood of the model, and c is the normalizing constant $c = mTn$, with m being the number of sequences, T the length of

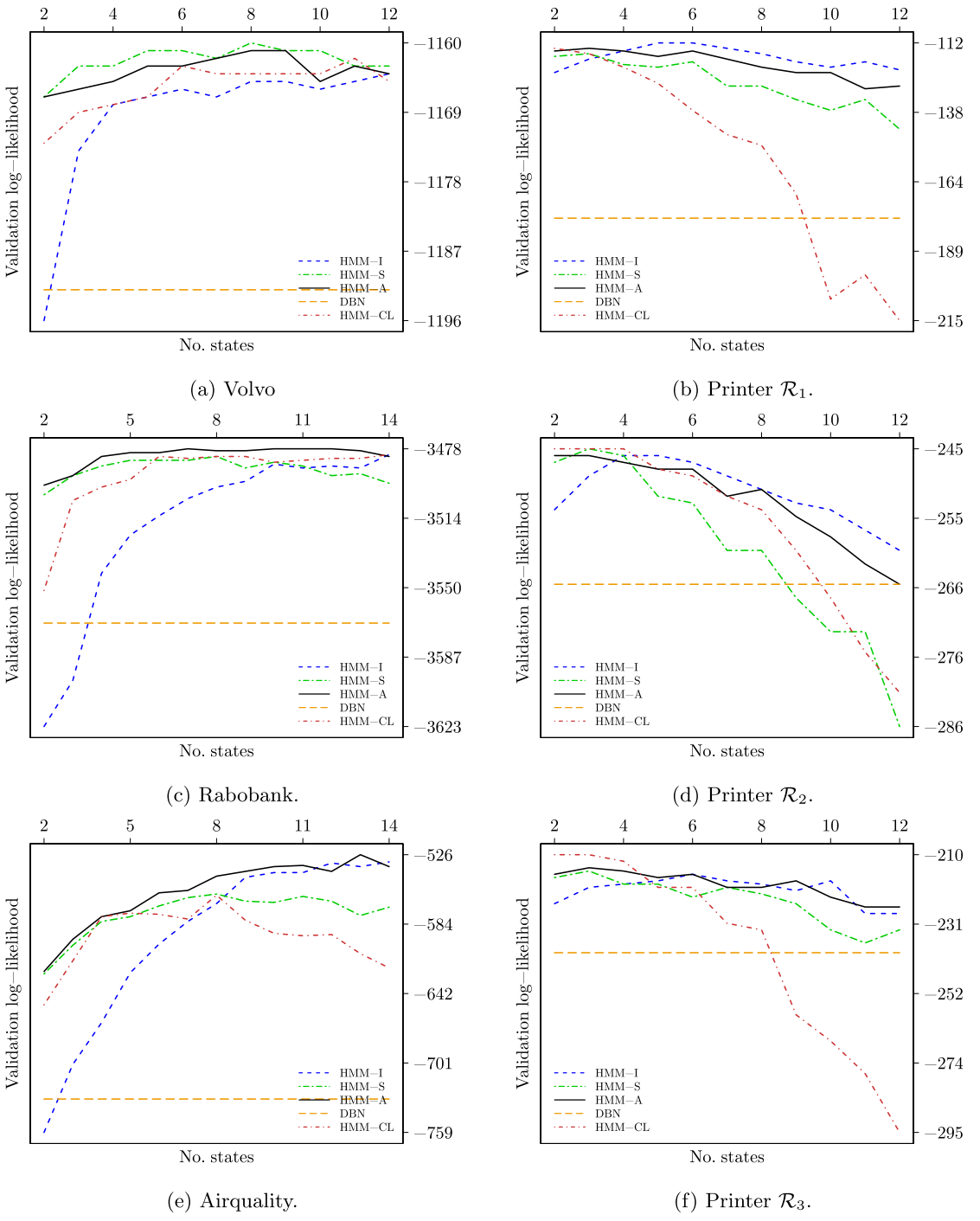


Fig. 11. Cross-validation log-likelihoods achieved by DBNs, symmetric, Chow–Liu, and asymmetric HMMs in real-world datasets. Each point represents the mean validation log-likelihood over 10 folds.

each sequence, and n the number of features in the dataset. As Table 5 shows, HMM-As generalized consistently better than symmetric HMMs and Chow–Liu HMMs. We further computed 95% confidence intervals (CIs) for these models as shown in Table 6, in order to assess the robustness of the generalization assessment (intervals were obtained by means of bootstrapping the testing datasets 2,000 times in each case). The CIs show that HMM-As could provide significantly better model quality on most scenarios of business process and Airquality cases. Significance in favor of HMM-As was also obtained in the printers cases, although HMM-As can be considered better than Chow–Liu HMMs in these cases but not significantly,

Table 6
95% bootstrap confidence intervals for the differences on generalization assessment (real-world datasets). Negative values indicate better fits for HMM-As. Notation: ** = HMM-A is significantly better; *A = HMM-A is better but not significantly; †X = model X is better but not significantly.

Dataset	HMM-A vs. HMM-I	HMM-A vs. HMM-S	HMM-A vs. HMM-CL
Volvo	**[−0.86, −0.11]****	[−0.14, 0.29]†S	**[−0.78, −0.21]****
Rabobank	[−0.15, 0.15]	[−0.47, 0.06]*A	[−0.02, 0.26]†CL
Airquality	**[−3.17, −0.17]****	[−3.08, 0.61]*A	**[−10.30, −1.77]****
Printer $\mathcal{R}_1$	[−10.35, 0.46]*A	[−7.88, 0.37]*A	[−1.32, 0.05]*A
Printer $\mathcal{R}_2$	[−1.53, 0.74]*A	**[−4.16, −0.38]****	[−1.58, 1.11]*A
Printer $\mathcal{R}_3$	**[−4.79, −0.3]****	[−1.67, 0.26]*A	[−3.35, 3.48]

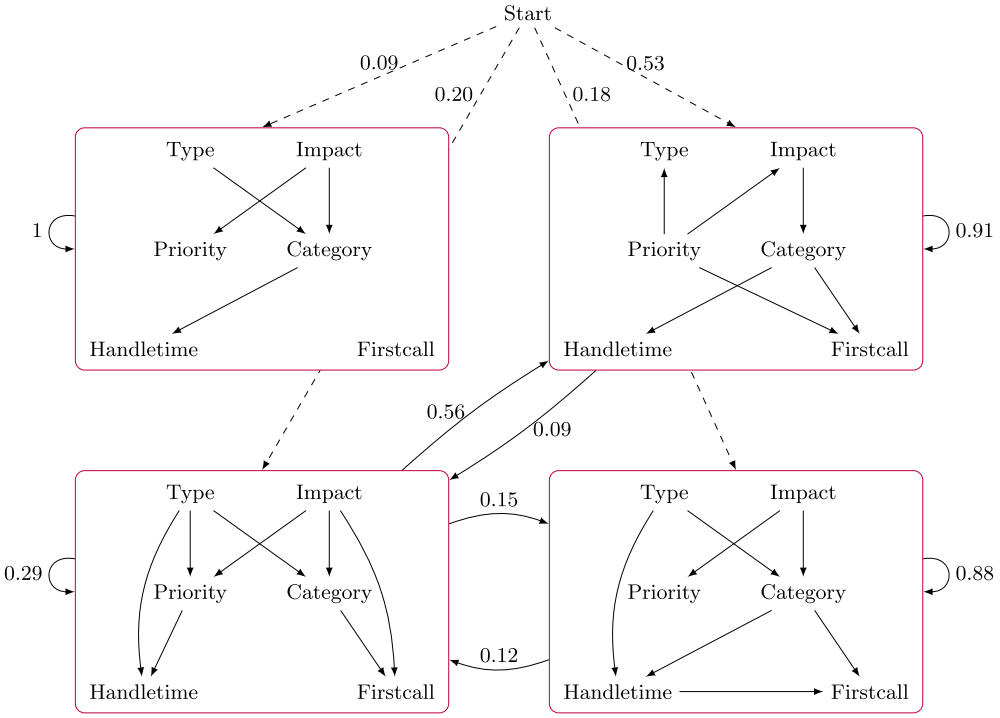


Fig. 12. HMM-A learned from Rabobank dataset. Dotted arcs indicate initial probabilities.

which is explained by the fact that the HMM-A states are all virtually associated to forests that are sparser than trees, as in Printer $\mathcal{R}_3$ (Fig. 16) and in Printers $\mathcal{R}_1$ and $\mathcal{R}_2$ as well [23].

From the results on real-data discussed in this section, it seems fair to conclude that not only more structure is beneficial for HMMs to better capture real-life problems, but also the *right* additional structure, as provided by HMM-As by their state-specific Bayesian-network distributions. The number of parameters in HMM-As was consistently lower than those of standard HMMs, independent HMMs and Chow–Liu HMMs, suggesting that diverse local structure exist and could be discovered by HMM-As.

6.3. Problem insight

We discuss in this section problem insight that can be gained from the learned HMM-As. We stress that from a fundamental perspective, where Bayesian networks are tools to facilitate reasoning with statistical (in)dependences, the fact that HMM-As can provide multiple graphical structures to explain how dynamic systems evolve over time (e.g. a business process) represents additional insight by its very nature. This contrasts to symmetric HMMs, where all those specificities are lost (or hidden across a number of CPTs at most), thus much less insight is likely to be gained.

6.3.1. Business process models

Fig. 12 shows the HMM-A learned from the Rabobank case (CPTs are not shown). The model shows that *Type* is unconditionally independent of *Impact* and *Priority* in the bottom right-most state, while this is not the case in the top right-most state. These structural information might be used, e.g., to further develop policies for scheduling different types of interactions at various moments: if the system is assumed to be in the bottom right-most state, a more flexible scheduling might be possible, where different types of interactions do not need to be handled by priority or impact, but instead could be han-

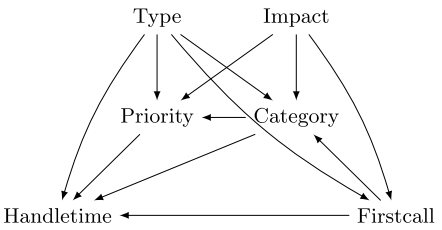


Fig. 13. Graphical structure of the emissions of the standard HMM learned from Rabobank dataset.

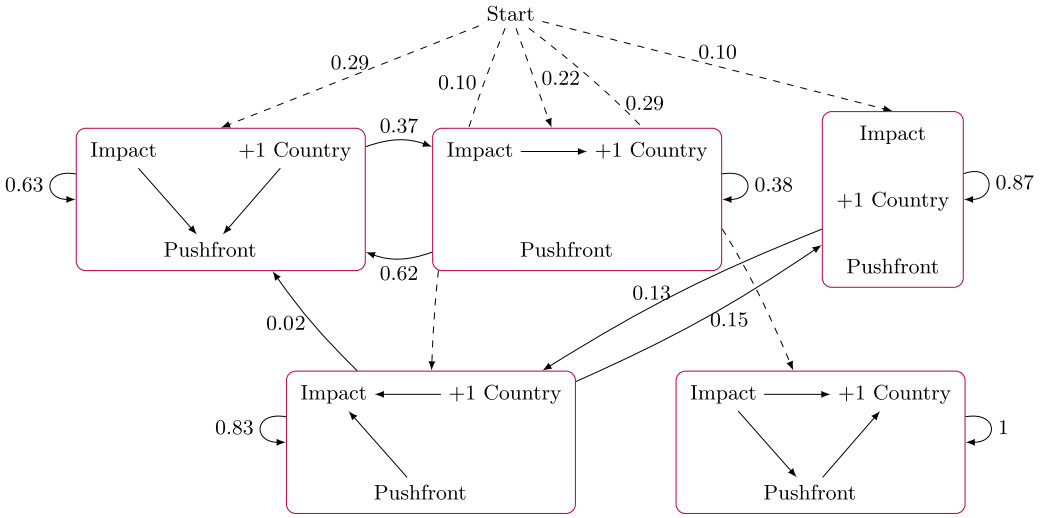


Fig. 14. HMM-A learned from Volvo dataset.

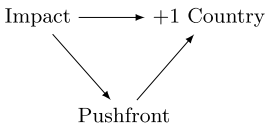


Fig. 15. Graphical structure of the emissions of the standard HMM learned from Volvo dataset.

dled by the expected time to be solved (due to the relationship with *Handletime*). On the other hand, if the system is in the bottom left-most state, *Type* is still unconditionally independent of *Impact*, but its unconditional independence of *Priority* no longer holds: in fact, such state seems to act as a bridge between the two aforementioned states. These information cannot be derived from the (almost fully connected) graphical structure of the learned HMM-S partially shown in Fig. 13, nor from the learned HMM-I. At a higher level of abstraction, HMM-As can also provide new insight by combining the local state properties with long-run behavior: this shows that batches of software-incident events that share independence properties alternate over time.

Fig. 14 shows the HMM-A learned from the Volvo dataset. As in the Rabobank case, this HMM-A is made of different graphical structures that lead to different independence relations, whereas the standard HMM has a fully connected structure as shown in Fig. 15. Finally, we note that in asymmetric models, not only probabilistic relations change, but also the structure in each state, providing evidence that these models capture an additional facet of the different stages the underlying dynamic system can transit to.

6.3.2. *Printers model*

Fig. 16 shows the HMM-A learned from Printer $\mathcal{R}_3$ dataset (we shall not discuss the other printers models as it has been done elsewhere [23]). This model suggests that the behavior of such large-scale printer alternates between two modes in the long run, which can be distinguished based on how the color rates C_1, C_2, C_3, and C_4 interact with the other observables. For example, once the printer is assumed to be in the right-most top state, one could decide whether the amount of maintenance performed could be altered in order to save resources, as this variable will not affect the colors' performance. However, this is probably not the case for most of the colors if the printer is in the center state, where those colors do interact with other observables. The standard HMM learned for this printer is shown in Fig. 17, lacking from such

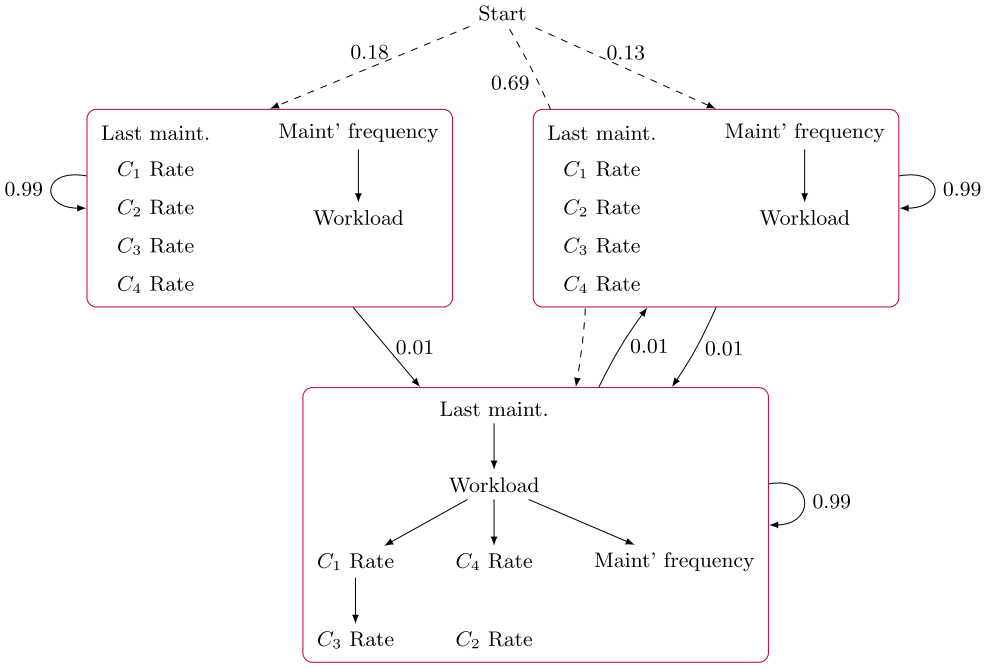


Fig. 16. HMM-A learned from Printer $\mathcal{R}_3$ case.

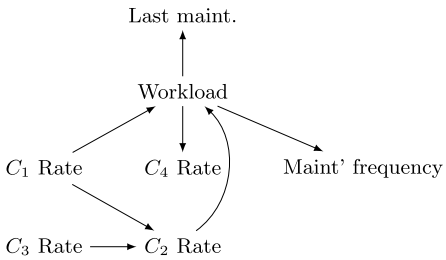


Fig. 17. Graphical structure of the emissions of the standard HMM learned from Printer $\mathcal{R}_3$ dataset.

specific alternation behavior that could be discovered by means of the HMM-A, as the colors variables are connected to all the other observables (whether directly or not).

7. Related work

Analyses of the sensitivity of Bayesian networks to parameter change are relatively numerous [45–47], however that does not seem to be the case when it comes to the sensitivity to the graphical structure. There is some research on how model structure affects accuracy in medical diagnosis problems [48], where the authors have shown that the accuracy was not significantly sensitive for disturbances on model structure, considering certain medical cases and diagnostic criteria. In the context of HMMs, however, the results shown in this paper suggest a different conclusion: with respect to model fit, modeling additional structure (by means of distribution asymmetries) seemed very important for achieving better model quality. Nevertheless, these conclusions are not necessarily contradictory in principle, as the employed criteria differ as well as the type of models. As in HMMs the state space dimension is a rather important parameter, modeling non-trivial structure that can lead to shorter state space seems crucial to such models, while this might not be the case to some static Bayesian networks. In fact, it is a general belief in the Bayesian networks field that non-trivial structure matters for better handling real-world problems [49–51].

In this paper, we attempted to provide a better understanding of the effects of modeling asymmetries on the feature space in HMMs. This involved a more thorough comparison of HMM-As with several families of HMMs, which is somewhat lacking in the literature. We considered not only independent HMMs, but also standard HMMs and Chow–Liu HMMs, the latter being able to model simpler asymmetries than HMM-As. Finally, the paper showed experiments involving models of DBNs, which are typically learned without hidden or latent variables.

8. Conclusions

In this paper, we investigated asymmetric hidden Markov models, which were presented in [23] in a preliminary form. HMM-As explicitly capture distribution asymmetries inherent to real-world problems, by means of associating individual hidden states to arbitrary Bayesian networks. We showed that, in principle, symmetric HMMs (e.g. independent and standard HMMs) can have their state space or emissions structure arbitrarily extended for representing structured distributions. Nevertheless, empirical results showed that this capability was not enough for guaranteeing comparable model quality, due to model overfitting (because of too many states or too dense emission structures). A similar conclusion holds for Chow–Liu HMMs, suggesting that going beyond tree-shaped asymmetries can be beneficial. In some real-world cases, adding structure (either via a symmetric or an asymmetric model) allowed for relevant model quality improvement, while the simplest model (i.e. the independent HMM) was good enough on others. This model selection issue could be adequately addressed by HMM-As, which provide enough flexibility to reduce the need for selecting a particular HMM architecture a priori.

Computationally, learning HMM-As introduce additional costs due to structure learning, compared to symmetric HMMs. Nevertheless, experiments indicated that good-quality HMM-As with compact state spaces could be obtained by using graphical structures found by common search heuristics. Hence, in practice learning HMM-As using structural EM resulted in fact in shorter running times compared to learning symmetric HMMs using standard EM in many cases. Furthermore, HMM-As learned from real-world datasets with varied sizes and number of observables were shown to bring additional problem insight that cannot be readily obtained from symmetric HMMs.

Several paths for further research can be considered. To some extent, HMM-As can be seen as tools for summarizing hidden Markov models with larger state spaces into models with more compact state spaces, as shown in the real-world experiments. We would like to further evaluate whether HMM-As act as *model summarizers* in more general settings, e.g. when the data generation mechanism has no explicit asymmetries (as in HMM-Is), and when it consists of different kinds of asymmetries (e.g. autoregressions, as in dynamic multinets). It could be also of interest to exploit the sensitivity of differences between state-specific networks, in the lines of works on sensitivity analysis for example. This could help eliminate too specific arcs that do not significantly contribute to model quality, thus allowing for more compact models. As we observed in real-world experiments and in simulations, several advantages obtained with HMM-As were more prominent when the problem had higher number of observables. Hence, we intend to further investigate such scaling aspect, as well as consider other real-world cases with more features and different types of observables (e.g. continuous and hybrid ones). Finally, we would like to compare the identification of asymmetries in sequential models as HMM-As with other approaches, such as knowledge compilation [52] and dynamic chain event graphs [53].

Acknowledgements

This work has been funded by NWO (Netherlands Organisation for Scientific Research), project Careful (62001863). We would like to thank Océ Technologies (Venlo, the Netherlands) for providing datasets of printers and aiding the discussion of results. We also thank the anonymous reviewers for their valuable comments that helped improving this paper.

Appendix A

Proof of Proposition 3.1. Let $\mathcal{M}$ be the given HMM-A and $\mathcal{M}'$ be a standard HMM, where both models are defined over $(\mathbf{X}, S)$. We construct $\mathcal{M}'$ for simulating $\mathcal{M}$ as follows. Let G_F be directed acyclic graph over $\mathbf{X}$ that is also fully connected. Add an arc from S to each $X_i \in \mathbf{X}$, and define the result as the graphical structure of the emissions of $\mathcal{M}'$. By the chain rule from probability theory, a fully connected structure can represent any probability distribution, hence, the distribution of each state in $\mathcal{M}$ can be represented by a state in $\mathcal{M}'$ by adequately parameterizing the CPTs on the emissions of $\mathcal{M}'$. This allows us to obtain $P'(\mathbf{X}^{(t)} \mid s^{(t)}) = P(\mathbf{X}^{(t)} \mid s^{(t)})$, for every state $s \in \mathrm{dom}(S)$.

Finally, we set the initial and transition distributions of $\mathcal{M}'$ to the same as those from $\mathcal{M}$. Thus, we conclude that $P'(\mathbf{X}^{(0:T)}, S^{(0:T)}) = P(\mathbf{X}^{(0:T)}, S^{(0:T)})$. □

Proof of Proposition 3.2. We construct in the following an independent HMM $\mathcal{M}'$ for simulating a given HMM-A $\mathcal{M}$. The observables are assumed to follow Bernoulli distributions each (an extension to multinomial distribution is straightforward). We denote by P and P' the joint distributions over $(\mathbf{X}^{(0:T)}, S^{(0:T)})$ of $\mathcal{M}$ and $\mathcal{M}'$ respectively.

Note that the state-specific BN associated to a state in $\mathcal{M}$ is a BN over n variables, hence its joint distribution can be completely characterized with at most $2^n - 1$ independent parameters, where we denote by $\theta_{\mathbf{x}}$ the parameter associated to the assignment $\mathbf{x}$. For each $\mathbf{x}$ and state s_i, we define a state $s_{i\mathbf{x}}$ in $\mathcal{M}'$. Moreover, we define emission parameters for each observable $X_j \in \mathbf{X}$:

$$P'(X_j = \top \mid s_{i\mathbf{x}}) \stackrel{\text{def}}{=} 1$$

whenever $(X_j = \top)$ holds in $\mathbf{x}$. After finishing this correspondence for all state-specific BNs of $\mathcal{M}$, we obtain $k2^n$ states in total in $\mathcal{M}'$. This finishes the construction of the emission distribution for $\mathcal{M}'$.

The remaining distributions of $\mathcal{M}'$ are constructed by taking the corresponding probabilities of $\mathcal{M}$ scaled by the emission probabilities of the joint assignments of $\mathbf{X}$ as follows. For the initial distribution, we define:

$$P'(s^{(0)}_{i\mathbf{x}}) \stackrel{\text{def}}{=} P(s^{(0)}_i)P(\mathbf{x} \mid s_i)$$

for each state s_i from $\mathcal{M}$ and assignment $\mathbf{x}$. On the other hand, we define the transitions in $\mathcal{M}'$ as:

$$P'(s^{(t+1)}_{j\mathbf{x}} \mid s^{(t)}_{i\mathbf{y}}) \stackrel{\text{def}}{=} P(s^{(t+1)}_j \mid s^{(t)}_i)P(\mathbf{x} \mid s_j)$$

It is straightforward to verify that this construction of $\mathcal{M}'$ produces valid probability distributions. Moreover, it assures that $P'(\mathbf{X}^{(0:T)}) = P(\mathbf{X}^{(0:T)})$. As a side note, observe that equality over $(\mathbf{X}^{(0:T)}, S^{(0:T)})$ does not hold in general. □

References

[1] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, K. Oura, Speech synthesis based on hidden Markov models, Proc. IEEE 101 (5) (2013) 1234–1252.
[2] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[3] K. Markov, J. Dang, S. Nakamura, Integration of articulatory and spectrum features based on the hybrid HMM/BN modeling framework, Speech Commun. 48 (2) (2006) 161–175.
[4] S.R. Eddy, Accelerated profile HMM searches, PLoS Comput. Biol. 7 (10) (2011) 1–16.
[5] M. Stanke, O. Schöffmann, B. Morgenstern, S. Waack, Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources, BMC Bioinform. 7 (2006) 62.
[6] D. Freitag, A. McCallum, Information extraction with HMM structures learned by stochastic optimization, in: Proc. of the 17th AAAI and 20th IAAI, AAAI Press, 2000, pp. 584–589.
[7] K. Seymore, A. Mccallum, R. Rosenfeld, Learning hidden Markov model structure for information extraction, in: AAAI 99 Workshop on Mach. Learning for Inf. Extr, 1999, pp. 37–42.
[8] A. Rozinat, M. Veloso, W. van der Aalst, Evaluating the quality of discovered process models, in: Proc. of Induction of Process Models, ECML PKDD, 2008, pp. 45–52.
[9] J. Bilmes, What HMMs can do, IEICE Trans. Inf. Syst. E 89-D (3) (2006) 869–891.
[10] Z. Ghahramani, An introduction to hidden Markov models and Bayesian networks, Int. J. Pattern Recognit. Artif. Intell. (2001) 9–42.
[11] Z. Ghahramani, M. Jordan, Factorial hidden Markov models, Mach. Learn. 29 (2) (1997) 245–273.
[12] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov model: analysis and applications, Mach. Learn. 32 (1) (1998) 41–62.
[13] A. Poritz, Linear predictive hidden Markov models and the speech signal, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 7, ICASSP '82, 1982, pp. 1291–1294.
[14] D. Geiger, D. Heckerman, Knowledge representation and inference in similarity networks and Bayesian multinets, Artif. Intell. 82 (1) (1996) 45–74.
[15] D. Heckerman, Probabilistic Similarity Networks, ACM Doctoral Dissertation Awards, MIT Press, 1991.
[16] J. Vlasselaer, W. Meert, G. van den Broeck, L. Raedt, Exploiting local and repeated structure in Dynamic Bayesian Networks, Artif. Intell. 232 (2016) 43–53.
[17] A. Cano, M. Gémez-Olmedo, S. Moral, Approximate inference in Bayesian networks using binary probability trees, Int. J. Approx. Reason. 52 (1) (2011) 49–62.
[18] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: Proc. of the 20th Intl. Conf. on Uncertainty in Artificial Intelligence, 1996, pp. 115–123.
[19] J. Pensar, H. Nyman, J. Lintusaari, J. Corander, The role of local partial independence in learning of Bayesian networks, Int. J. Approx. Reason. 69 (2016) 91–105.
[20] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: Proc. of the 20th UAI, UAI'96, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996, pp. 252–262.
[21] S. Kirshner, P. Smyth, A. Robertson, Conditional Chow–Liu tree structures for modeling discrete-valued vector time series, in: Proc. of the 20th UAI, 2004, pp. 317–324.
[22] J. Bilmes, Dynamic Bayesian multinets, in: Proc. of the Sixteenth Conference on Uncertainty in Artificial Intelligence, 2000, pp. 38–45.
[23] M.L.P. Bueno, A. Hommersom, P.J.F. Lucas, S. Verwer, A. Linard, Learning complex uncertain states changes via asymmetric hidden Markov models: an industrial case, in: Proc. of the 8th Int. Conf. on Probabilistic Graphical Models, Lugano, 2016, pp. 50–61.
[24] N. Friedman, Learning belief networks in the presence of missing values and hidden variables, in: Proc. of the 14th ICML, ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, USA, 1997, pp. 125–133.
[25] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc., Ser. B, Stat. Methodol. 39 (1) (1977) 1–38.
[26] K. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. Thesis, UC Berkeley, Computer Science Division, Jul. 2002.
[27] Y. Bengio, P. Frasconi, An input output HMM architecture, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, vol. 7, The MIT Press, 1995, pp. 427–434.
[28] A. Motzek, R. Möller, Indirect causes in dynamic Bayesian networks revisited, in: Proc. of the 24th Intl. Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 2015, pp. 703–709.
[29] B.-H. Juang, L. Rabiner, Mixture autoregressive hidden Markov models for speech signals, IEEE Trans. Acoust. Speech Signal Process. 33 (6) (1985) 1404–1413.
[30] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
[31] D.M. Chickering, Learning Bayesian Networks Is NP-Complete, Springer New York, New York, NY, 1996, pp. 121–130.
[32] J. Bilmes, A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Tech. Rep. TR-97-021, International Computer Science Institute, Berkeley, 1998.
[33] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.
[34] P. Larrañaga, M. Poza, Y. Yurramendi, R.H. Murga, C.M.H. Kuijpers, Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters, IEEE Trans. Pattern Anal. Mach. Intell. 18 (9) (1996) 912–926.
[35] D. Margaritis, Learning Bayesian Network Model Structure from Data, Ph.D. Thesis, Carnegie Mellon University, 2003.
[36] J. Cussens, M. Järvisalo, J.H. Korhonen, M. Bartlett, Bayesian network structure learning with integer programming: polytopes, facets and complexity, J. Artif. Intell. Res. 58 (2017) 185–229.
[37] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[38] M. Scutari, J. Denis, Bayesian Networks with Examples in R, Chapman and Hall, Boca Raton, 2014.
[39] M.R.G. Bela, A. Frigyik, Amol Kapila, Introduction to the Dirichlet Distribution and Related Processes, Technical Report UWEETR-2010-0006, Department of Electrical Engineering, University of Washington, 2010, 28 pages.
[40] G. Melançon, F. Philippe, Generating connected acyclic digraphs uniformly at random, Inf. Process. Lett. 90 (4) (2004) 209–213.
[41] W. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer Publishing Company, Incorporated, 2011.
[42] W. Steeman, BPI challenge 2013, incidents, http://dx.doi.org/10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee, 2013.
[43] B. van Dongen, BPI challenge 2014: interaction details, http://dx.doi.org/10.4121/uuid:3d5ae0ce-198c-4b5c-b0f9-60d3035d07bf, 2014.
[44] S.D. Vito, M. Piga, L. Martinotto, G.D. Francia, CO, NO_2 and NO_x urban pollution monitoring with on-field calibrated electronic nose by automatic Bayesian regularization, Sens. Actuators B, Chem. 143 (1) (2009) 182–191.
[45] A. Oniśko, M.J. Druzdzel, Impact of precision of Bayesian network parameters on accuracy of medical diagnostic systems, Artif. Intell. Med. 57 (3) (2013) 197–206.
[46] L.C. van der Gaag, S. Renooij, V.M. Coupé, Sensitivity Analysis of Probabilistic Networks, Springer, Berlin, Heidelberg, 2007, pp. 103–124.
[47] E. Castillo, J.M. Gutierrez, A.S. Hadi, Sensitivity analysis in discrete Bayesian networks, IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. 27 (4) (1997) 412–423.
[48] A. Oniśko, M.J. Druzdzel, Impact of Bayesian Network Model Structure on the Accuracy of Medical Diagnostic Systems, Springer International Publishing, Cham, 2014, pp. 167–178.
[49] R. Daly, Q. Shen, S. Aitken, Review: learning Bayesian networks: approaches and issues, Knowl. Eng. Rev. 26 (2) (2011) 99–157.
[50] N. Friedman, The Bayesian structural EM algorithm, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 129–138.
[51] M. Scanagatta, C.P. de Campos, G. Corani, M. Zaffalon, Learning Bayesian networks with thousands of variables, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., 2015, pp. 1864–1872.
[52] M. Chavira, A. Darwiche, On probabilistic inference by weighted model counting, Artif. Intell. 172 (6) (2008) 772–799.
[53] L.M. Barclay, R.A. Collazo, J.Q. Smith, P.A. Thwaites, A.E. Nicholson, The dynamic chain event graph, Electron. J. Stat. 9 (2) (2015) 2130–2169.