Classification Among Hidden Markov Models

S. Akshay
IIT Bombay, India

Hugo Bazille
Univ Rennes, Inria, IRISA, France

Eric Fabre
Univ Rennes, Inria, IRISA, France

Blaise Genest
Univ Rennes, CNRS, IRISA, France

Abstract

An important task in AI is one of classifying an observation as belonging to one class among several (e.g. image classification). We revisit this problem in a verification context: given k partially observable systems modeled as Hidden Markov Models (also called labeled Markov chains), and an execution of one of them, can we eventually classify which system performed this execution, just by looking at its observations? Interestingly, this problem generalizes several problems in verification and control, such as fault diagnosis and opacity. Also, classification has strong connections with different notions of distances between stochastic models.

In this paper, we study a general and practical notion of classifiers, namely *limit-sure classifiers*, which allow misclassification, i.e. errors in classification, as long as the probability of misclassification tends to 0 as the length of the observation grows. To study the complexity of several notions of classification, we develop techniques based on a simple but powerful notion of stationary distributions for HMMs. We prove that one cannot classify among HMMs iff there is a *finite separating word* from their stationary distributions. This provides a direct proof that classifiability can be checked in PTIME, as an alternative to existing proofs using *separating events* (i.e. *sets of* infinite separating words) for the total variation distance. Our approach also allows us to introduce and tackle new notions of classifiability which are applicable in a security context.

2012 ACM Subject Classification Theory of computation

Keywords and phrases verification: probabilistic systems, partially observable systems

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2019.29

Related Version A full version of the paper is available at `http://perso.crans.org/genest/abfg19.pdf`.

Funding This work has been partially supported by DST/CEFIPRA/INRIA Associated team EQuaVE and DST Inspire Faculty Award [IFA12-MA-17].

Acknowledgements We would like to thank Stefan Kiefer for his expert opinion, as well as anonymous reviewers for their constructive comments.

1 Introduction

The spectacular success of artificial intelligence (AI) and machine learning techniques in many varied application domains in the last decade has led to the emergence of several new and old questions, especially regarding their guarantees and correctness. This has led to several recent projects at the interface of formal methods and AI, whose broad goal is to formally reason and verify properties about these AI models and tasks. One such important task in AI is classification, which is a fundamental problem with many practical applications,

39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019).
Editors: Arkadev Chattopadhyay and Paul Gastin; Article No. 29; pp. 29:1–29:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

e.g., in image processing. In this paper, we consider classification in a verification context. One main issue when verifying systems is partial observability. It is thus important to know what information can be recovered from a partially observable system.

We first consider a system perspective: we want to know whether, no matter the execution of the system, some hidden information is retrievable, at least with high probability. To represent the system, we thus consider a partially observable stochastic model, namely *Hidden Markov Models (HMM for short)* [12, 10], also known as labeled Markov chains [6] or probabilistic labeled transition systems [5]. While notationally different, these various models are equivalent in terms of expressive power. In HMMs, states are not observable, but we get some (potentially stochastic) signals from states. In the specific variant of HMMs that we study in this paper, we encode the signals from states as labels of transitions exiting states. That is, the observation from an execution of an HMM is its labeling sequence. We encode the different hidden information as several HMMs, with different transition probabilities. Finding the hidden information from the observation thus amounts to classifying the observation among the different HMMs.

Many problems concerning systems with hidden information can be recast in the framework of classification, such as, (i) fault diagnosis: classifying between a faulty system that has executed errors and the system without faults [14, 15, 4, 5]; (ii) opacity: classifying between high and low privilege parts of the system [10], etc. Although some problems are incomparable (e.g. diagnosis is intrinsically “asymmetric” while classification is “symmetric”), most proof techniques and ideas are common. Moreover, results on classification problems have been applied to show results in these related contexts. While it is not our aim to survey these applications here, we provide two instances: a fault diagnosis problem [5] is solved using a result on distance between stochastic systems [6], which is equivalent with classification [11]. Also, opacity is cast as a classification problem in [10]. We hence believe that classification is a good framework to state and prove algorithmic and complexity results.

Several notions of classification can be defined: sure, almost-sure, and limit-sure, depending respectively on whether we want the classification to eventually happen for sure, with probability 1, or with arbitrarily small error. The first two notions have classical solutions coming from fault-diagnosis [14, 4]: the existence of such classifiers can be checked in PTIME and PSPACE respectively. The third notion is however the most practical as the classifier is the most powerful: it can use the long run statistics on observations to take its decision (e.g. the frequency of ab’s in the word). It is also the hardest notion to study for this very reason.

We focus on this notion of limit-sure classification in this paper. First, a closely-related problem of *distinguishability* has been proved to be in PTIME by [11], using the PTIME algorithm from [6] to test whether the total variation metric between two HMMs is 1. We reinvestigate these deep results using different techniques, which shines some new light on this problem. Our starting point is the following: for a very restricted class of HMMs [10], whose underlying Markov chains are ergodic and crucially, assuming that initial distributions have non-zero probability on every state, it is sufficient to consider the statistics on *states* (e.g. the frequency of state s). These statistics on *states* are obtained by [10] using the classical notion of stationary distributions over the underlying Markov Chain, i.e. the HMM where we forget all observations. As we show in Example 2, stationary distributions on Markov chains do not suffice for solving limit-sure classification for general HMMs. We build on this idea and propose a new notion to study the long run statistics of the *observations.*

Our first contribution is to develop the notion of stationary distributions for *general HMMs* to study the long run statistics of the observations. To do so, we focus on beliefs, that is, the set of states that can be reached with the same observation. We show that a notion of stationary distributions can be defined for beliefs in Bottom Strongly Connected Components (BSCCs), and that it also corresponds to a notion of asymptotic distributions, describing the asymptotic statistics of beliefs. This generalizes stationary distributions for Markov chains: for instance, irreducible Markov chains of period k correspond to cycling through k different beliefs. We believe that this notion can find applications in other contexts.

Our next contribution is to show how this notion of stationary distribution of HMMs can be used to characterize limit-sure classifiability. We show that we cannot classify between HMMs iff they have beliefs which can be reached by the same observation and for which the stationary distributions can be separated by *one finite word* (for which the probability is different). This provides a PTIME algorithm to test for limit-sure classifiability. Note that the existence of such a PTIME algorithm has been established in [11], where this result was formulated in terms of HMMs distinguishability. The proofs are different however, as [11] focuses on separating events [6], that is *sets of infinite words* with probability 0 (resp. 1) in one of the HMMs (resp. the other one), while considering stationary distributions allows us to focus on a single finite separating word with probability p (resp. $q \neq p$).

Our final contribution is to study classifiability in a security context: an attacker has different attacks against different HMMs. To be able to perform his attack, he needs to find one execution that can be classified (and thus attacked) rather than whether every execution can be classified. We call this notion *attack-classification*. We study limit-sure attack-classification using the notion of stationary distributions for HMMs developed above. We show that deciding whether there exists a limit-sure attack-classifier between two HMMs is PSPACE-complete. On the other hand, if we consider a variation on the notion of limit-sure attack-classifier, which extends distinguishability for HMMs [11], we are able to show that it is not only different from limit-sure attack-classifier, but this problem is also undecidable. All missing proofs and details can be found in the long version [1].

2 Preliminaries and Problem Statement

A *Hidden Markov Model* [12, 13, 10] (*HMM* for short) $\mathcal{A}$ on finite alphabet Σ is a tuple $\mathcal{A} = (S, M, \sigma_0)$ with S a set of states, σ_0 an initial distribution, $M : S \times \Sigma \times S \to [0; 1]$, such that for all s, $\sum_{a,s'} M(s, a, s') = 1$. Notice that this notion has been referred to using different names in the literature: labeled Markov chains, pLTS (probabilistic transition systems) in [5], probabilistic automata (not to be confused with Rabin's Probabilistic automata), etc. Classical Markov chains can be viewed as HMMs with a single letter alphabet. In what follows we assume knowledge of classical properties, definitions about Markov chains, such as irreducibility, aperiodicity and refer to [9] for a formal treatment.

A run ρ of $\mathcal{A}$ is a sequence in $S(\Sigma \times S)^*$. It starts in $s^-(\rho)$, with $\sigma_0(s^-(\rho)) > 0$, and ends in state $s^+(\rho)$. An observation w from $\mathcal{A}$ is a sequence of letters $w = a_1 \cdots a_n \in \Sigma^*$ such that there exists a run ρ made of $n+1$ states $\rho = s_0, a_1 \ldots, a_n s_n$ with $\sigma_0(s_0) > 0$ and for all $i > 0$, $M(s_{i-1}, a_i, s_i) > 0$. We denote $obs(\rho) = w$. For a run $\rho = s_0, a_1 \ldots, a_n s_n$, we define its probability as $P(\rho) = \sigma_0(s_0) \cdot \prod_{i=1}^n M(s_{i-1}, a_i, s_i)$. We sometimes abuse notation and write $M(s_1, w, s_n)$ to mean $\prod_{i=1}^n M(s_{i-1}, a_i, s_i)$. We define the probability of an observation $w \in \Sigma^*$ as $P(w) = \sum_{\rho | obs(\rho)=w} P(\rho)$. In general we write $P_\sigma^{\mathcal{A}}$ to express the probability in HMM $\mathcal{A}$ with initial distribution σ. If $\sigma(s) = 1$, then we use $P_s^{\mathcal{A}}$ instead.

A *non-deterministic finite automaton (NFA for short)* is as usual a structure $\mathcal{A} = (S, \Delta, S_0)$, where the transition probabilities (as in a HMM) are replaced with a transition relation Δ and initial distribution is replaced by a set of initial states S_0. For a HMM (S, M, σ_0), we can associate the NFA $\mathcal{A} = (S, \Delta, S_0)$, by taking $(s, a, t) \in \Delta$ iff $M(s, a, t) > 0$ and $s \in S_0$ iff $\sigma_0(s) > 0$. The notion of paths and observation is preserved. Fig. 1 shows an HMM on the left and an NFA on the right.

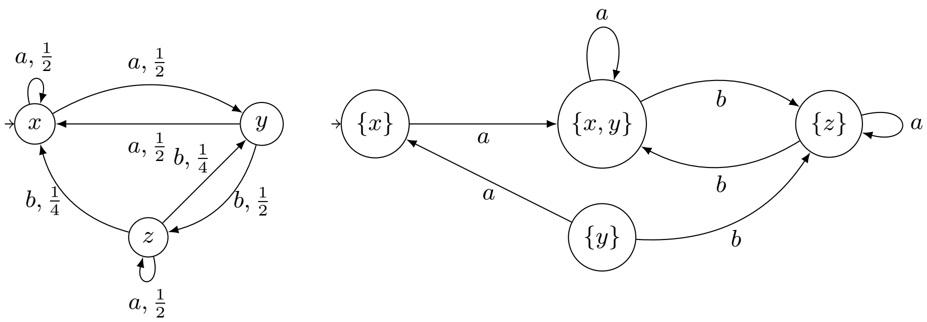

Figure 1 Example of an HMM $\mathcal{A}$ on alphabet $\Sigma = \{a, b\}$ and of an NFA $\mathcal{B}_{\mathcal{A}}$ on alphabet Σ.

The language of an automaton (or by extension of an HMM) is the set of observations $L(\mathcal{A}) = \{w \mid w = obs(\rho), \rho \text{ a path of } \mathcal{A}\}$. We denote by $L^\infty(\mathcal{A})$ the set of infinite observations in $\mathcal{A}$, that is such that every of its prefix is in $L(\mathcal{A})$. Finally, we use the standard way to extend probabilities to some sets of infinite paths, by means of cylinder-sets [2]. In particular, taking two HMMs $\mathcal{A}_1, \mathcal{A}_2$ on the same alphabet, $L^\infty(\mathcal{A}_1) \cap L^\infty(\mathcal{A}_2)$ is measurable. We write $L(\mathcal{A}, s)$ for the language of $\mathcal{A}$ starting in state s.

A strongly connected component C of an HMM $\mathcal{A}$ is a maximal set of states such that there is a path from any state of C to any state of C. A strongly connected component C is called a *bottom strongly connected component(BSCC)* if the only states reachable from C are in C. For instance, there is only one BSCC in the NFA of Fig. 1, with 2 states $\{x, y\}$ and $\{z\}$. Runs of an HMM end up in one of the BSCCs with probability 1.

Probabilistic Finite Automata (PFA). Several lower bounds will come from results on Rabin's *probabilistic finite automaton (PFA)* [8]. A PFA $\mathcal{A}$ on finite alphabet Σ is a tuple $\mathcal{A} = (S, (M_a)_{a \in \Sigma}, \sigma_0)$ with S a set of states, σ_0 an initial distribution, $M_a : S \times S \to [0, 1]$ for each $a \in \Sigma$, such that for all a, s, $\sum_{s'} M_a(s, s') = 1$. Similar to HMMs, the states of a PFA are not observed, but only letters $a \in \Sigma$ are. The difference is that we can control a PFA by choosing an action $a \in \Sigma$, while in HMMs, we observe passively an observation $a \in \Sigma$.

2.1 Probabilistic equivalence can be checked in PTIME

The PTIME algorithm for probabilistic equivalence is at the core of the PTIME algorithms from [6] (and hence [11, 5] using it), [10] and ours. Let σ_1, σ_2 be distributions over states of HMMs $\mathcal{A}_1, \mathcal{A}_2$ respectively. HMMs $\mathcal{A}_1, \mathcal{A}_2$ are equivalent from distributions σ_1, σ_2, denoted $(\mathcal{A}_1, \sigma_1) \equiv (\mathcal{A}_2, \sigma_2)$, if for any observation $w \in \Sigma^*$, we have $P^{\mathcal{A}_1}_{\sigma_1}(w) = P^{\mathcal{A}_2}_{\sigma_2}(w)$. In [3] (see also [6]), it is shown how to test in polynomial time whether $P^{\mathcal{A}_1}_{\sigma_1} \equiv P^{\mathcal{A}_2}_{\sigma_2}$, i.e.

$$\forall w \in \Sigma^*, \quad (\sigma_1 \quad \sigma_2) \cdot \begin{bmatrix} M_1(w) & \emptyset \\ \emptyset & M_2(w) \end{bmatrix} \cdot (1, \cdots, 1, -1, \cdots, -1)^T = 0$$

As the dimension of $Eq(\mathcal{A}_1, \mathcal{A}_2) = \{\begin{bmatrix} M_1(w) & \emptyset \\ \emptyset & M_2(w) \end{bmatrix} \cdot (1, \cdots, 1, -1, \cdots, -1)^T \mid w \in \Sigma^*\}$ is at most $|\mathcal{A}_1| + |\mathcal{A}_2|$, we can build a basis $v_1, \dots v_\ell$ for $Eq(\mathcal{A}_1, \mathcal{A}_2)$ of size $\ell \leq |\mathcal{A}_1| + |\mathcal{A}_2|$. It suffices then to check whether $(\sigma_1 \quad \sigma_2) \cdot v_i = 0$ for all $i \leq \ell$.

Notice that equivalence of PFA has been known to be in PTIME for 30 years [16], before HMMs [3]. Actually, equivalences of HMMs of and PFAs are inter-reducible (one direction can be found in [7], and the other one is easy by considering the HMM associated with a PFA, which performs actions of the PFA uniformly at random).

2.2 The classification problem and its variants

Let $(A_i)_{i \leq k}$ be a set of HMMs representing different behaviors of a system under observation. The system secretly picks one HMM behavior to follow, i.e. it is a priori unknown which of the HMMs is being followed by the system. We want to *classify*, i.e. find out, which HMM behavior the system follows, only by looking at the observation $w \in \Sigma^*$. The longer we observe the system, the larger the length of the observation and better the information we have to find out the HMM. This leads us to the notion of classifiability. As it suffices to consider HMMs pairwise, we will consider in the following there is only a choice between $k = 2$ HMMs. We will denote them by $\mathcal{A}_1$, with n states, and $\mathcal{A}_2$, with m states. Formally, a classifier is a function $f : \Sigma^* \to \{\perp, 1, 2\}$ that outputs the index of the HMM from an observation, or possibly $\perp$ if it cannot conclude (yet). Consider for example $\mathcal{A}_1, \mathcal{A}_2$, both following the HMM in Figure 1, the difference being that $\mathcal{A}_1$ starts in x while $\mathcal{A}_2$ starts in z. If the observation starts with b, then we know the systems follows $\mathcal{A}_2$, because b is not possible from x. We can thus let $f(bw) = 2$. However, if the observation is ab^2a, then it could come from any $\mathcal{A}_1$ or $\mathcal{A}_2$. If the systems are probabilistically equivalent, then no matter how much we observe, we cannot classify among them. However, this is one extreme case. One can consider several notions of classifiability:

- *sure classifiability*: there exists a classifier f that eventually identifies the accurate HMM that generated w. That is, for all $w \in \Sigma^\infty$, there exists a finite prefix v of w and a classifier f for v such that $f(v) = 1$ (resp. $f(v) = 2$) iff there is no path ρ of $\mathcal{A}_2$ (resp. of $\mathcal{A}_1$) with $obs(\rho) = w$.
- *almost-sure classifiability*: there exists a classifier f that eventually identifies the accurate HMM that generated w with probability 1. This classifier cannot do errors, but there may be some infinite observation that cannot be classified, though the probability it happens should be 0 (such as tossing tail forever on a fair coin).
- *limit-sure classifiability*: there exists a classifier f that, for all $\epsilon > 0$, eventually provides the accurate HMM with probability $> 1 - \epsilon$. This is the most general notion: sure implies almost-sure implies limit-sure classifiability.

This leads to the two main questions that we are interested in, for each of the above notions: (i) how easy is it to decide if there exists a classifier? (ii) if there exists a classifier, how easy is it to produce one explicitly? For the first question, we can answer easily for the two first notions, which have been studied in different contexts.

▶ **Proposition 1** ([14, 4]). *We can surely classify among 2 HMMs iff $L^\infty(\mathcal{A}_1) \cap L^\infty(\mathcal{A}_2) = \emptyset$, and this can be checked in PTIME. We can almost-surely classify among 2 HMMs iff the set $L^\infty(\mathcal{A}_1) \cap L^\infty(\mathcal{A}_2)$ has probability 0, and this is a PSPACE-complete problem.*

For the first two notions, building the classifier is also easy: intuitively, it suffices to compute the set of states reached with the observation (called *belief* in the next section) for both HMMs. If the system is classifiable, one of these sets will eventually (almost surely with the second notion) become empty. The classifier answers the HMM with non-empty set.

Unlike the two first notions, *limit-sure classifiability* cannot be expressed in terms of the language. Indeed, it is possible to limit-surely classify among $\mathcal{A}_1, \mathcal{A}_2$, and yet $L(\mathcal{A}_1) = L(\mathcal{A}_2)$. Also, a limit-sure classifier can use statistics in order to give its estimate, which opens a lot of possibilities. Let us illustrate these:

▶ **Example 2.** Consider again $\mathcal{A}_1, \mathcal{A}_2$, where both are the HMM $\mathcal{A}$ from Fig. 1, where $\mathcal{A}_1$ starts from state x and $\mathcal{A}_2$ starts from state z. If the observation starts with b, then it is easy to conclude that the HMM is $\mathcal{A}_2$. If it starts with a, then the set of states which can be reached after observation a is $\{x, y\}$ in $\mathcal{A}_1$ and $\{z\}$ in $\mathcal{A}_2$, which are both in the BSCCs. Actually, after an even number of b's (and any number of a's), we still have $\{x, y\}$ the set of states possible in $\mathcal{A}_1$ and $\{z\}$ in $\mathcal{A}_2$. In the following section using stationary distributions on HMMs, we will show how to compute that if the HMM is $\mathcal{A}_1$, after an even number of b's, the long term average is $\frac{3}{5}$ to be in x and $\frac{2}{5}$ to be in y. From this, we deduce that the long term average is $\frac{4}{5} = \frac{3}{5}1 + \frac{2}{5}\frac{1}{2}$ to perform an a after an even number of b's. On the other hand, if the HMM is $\mathcal{A}_2$, then the state is z and we obtain the long term average $\frac{1}{2}$ to perform letter a after an even number of b's. As the observation grows, the average frequency over the observation will tend towards the long term average by law of large numbers. Thus the classifier $f(w) = 1$, if the average frequency of a's after an even number of b's observed in w is closer to $\frac{4}{5}$ than to $\frac{1}{2}$, is limit-sure. Notice that using the standard stationary distributions on *Markov chains* as in [10] only tells us that both $\mathcal{A}_1$ and $\mathcal{A}_2$ stay in long term average frequency $\frac{3}{7}$ in x, $\frac{2}{7}$ in y, and $\frac{2}{7}$ in z , and thus do $\frac{5}{7} = \frac{3}{7} + \frac{2}{7}\frac{1}{2} + \frac{2}{7}\frac{1}{2}$ of a's in average, which cannot limit-surely classify between $\mathcal{A}_1, \mathcal{A}_2$.

From the point of view of practical applicability, limit-sure classifiers are the most powerful, although harder to study. In Section 4, we will study limit-sure classifiability, that we simply call classifiability. In Section 5, we further generalize this notion to a game-theoretic attack-classification framework, which is applicable in security settings.

3 Stationary distributions for HMMs

In order to solve limit-sure classification, we would like to use statistics on observations. Stationary distributions, which is a concept developed for *Markov chains*, tells us the frequency to expect about states, as used in [10]. We generalize this concept to HMMs to take into account observations. While stationary distributions for HMMs turn out to be crucial in the realm of classifiability, we believe it is also of independent interest.

For a Markov chain M, a stationary distribution σ is a distribution over states of M such that $\sigma \cdot M = \sigma$. In HMMs, the observation plays an important role and changes our knowledge of states in which the run could be. Thus, we consider the set of states that could be reached in an HMM $\mathcal{A}$ with a given observation, and call this as the *belief-state* or just *belief*. Formally, let w be an observation. The *belief* $B_{\mathcal{A}}(w)$ associated with w is the set of states $\{s^+(\rho) \mid obs(\rho) = w\}$ which can be reached by a path labeled by w. For instance, with the HMM $\mathcal{A}$ from Fig. 1, we have $B_{\mathcal{A}}(aa) = \{x, y\}$. We let $\mathcal{B}_{\mathcal{A}} = (2^S, \Delta, \mathbf{s_0})$ be the *belief automaton* associated with $\mathcal{A}$: (i) its states represent the beliefs associated with observations of $\mathcal{A}$, (ii) we have $(B, a, B') \in \Delta$ if $B' = \{s' \mid \exists s \in B, M(s, a, s) > 0\}$, and (iii) $\mathbf{s_0} = \{s \mid \sigma_0(s) > 0\} \in 2^S$. This is the usual subset construction used for determinizing an automaton, as shown on Fig. 1. As $\mathcal{B}_{\mathcal{A}}$ is deterministic, we sometimes abuse notation and denote $\Delta(B, a)$ for the unique B' with $(B, a, B') \in \Delta$.

Consider a BSCC D of HMM $\mathcal{A}$ (as for Markov chains, this is to ensure irreducibility). For $x \in D$, we denote by $\mathcal{B}^x_D$ the subgraph of $\mathcal{B}_{\mathcal{A}}$ reachable from $\{x\}$. On figure 1, we have $\mathcal{B}^y_D = \mathcal{B}_{\mathcal{A}}$. It has a unique BSCC, with 2 beliefs $\{x, y\}$ and $\{z\}$. We now show that this is the general form of the belief automaton:

▶ **Lemma 3.** *There is a unique BSCC in $\mathcal{B}^x_D$, and it does not depend upon $x \in D$.*

We denote E_D the set of beliefs X in the unique BSCC of $\mathcal{B}^x_D$, and $E_{\mathcal{A}}$ the union over all BSCCs D of $\mathcal{A}$. Notice that $E_{\mathcal{A}}$ may not contain all beliefs in the BSCCs of $\mathcal{B}_{\mathcal{A}}$, because we restrict ourselves to beliefs X reachable from $\{x\}$ with a single state x of a BSCC of $\mathcal{A}$. This is crucial for Lemma 3 to hold. We will see that considering singletons is not a

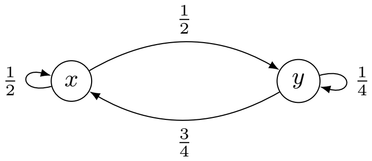

Figure 2 Markov chain $M_{x,y}$ associated with the belief $\{x, y\}$.

restriction: assume that the belief reached in a BSCC of beliefs comes from two different states x, y. Either the statistics on observation from x and y are the same, in which case we change nothing by considering them only from x. Otherwise, they have different statistics on observation, and looking at the observed statistics will give away with arbitrarily small error the state x or y which they originate from.

For Markov chains (i.e. HMMs on a one letter alphabet), the BSCC E_D is exactly $X_1 \to X_2 \cdots \to X_k \to X_1$, with k the *period* of this BSCC. Hence, this construction can be seen as a generalization to HMMs of the notion of period of a Markov chain. We use it to generalize the Fundamental theorem of Markov chains to HMMs.

Let $X \in E_{\mathcal{A}}$. We are interested in the *asymptotic distribution* associated to belief X, that is the statistics over states of X given that the belief state is X. From that, we will be able to deduce the statistics over observations. Let W_X the (possibly countable infinite) set of words which brings from belief X to belief X without seeing belief X in-between. Consider $\sigma_{y,i}$ the distribution over X such that $\sigma_{y,i}(x) = \sum_{w \in W_X^i} M(y, w, x)$, the probability of reaching x from y after seeing i words of W_X. To compute the limit of $\sigma_{y,i}$, we define the stationary distribution $\sigma_X : X \to [0, 1]$ of the HMM given a belief X. For that, we enrich the states of $\mathcal{A}$ with its beliefs, considering the product $\mathcal{A} \times \mathcal{B}_{\mathcal{A}}$ (same runs with same probabilities as in $\mathcal{A}$). For all $x, y \in X$, let $M_X(x, y)$ be the probability in the HMM $\mathcal{A} \times \mathcal{B}_{\mathcal{A}}$ to reach (y, X) from (x, X) before reaching any other (z, X), $z \neq y$ (we refer to [2] to compute $M_X(x, y)$ for all x, y). We have that for all $x \in X$, $\sum_{y \in X} M_X(x, y) = 1$, that is M_X is a Markov chain. For instance, on Fig. 1, let $X = \{x, y\} \in E$. The Markov chain M_X is depicted in Fig. 2 has a unique stationary distribution $\sigma(x) = \frac{3}{5}$ and $\sigma(y) = \frac{2}{5}$. We obtain:

▶ **Theorem 4.** *Given a HMM $\mathcal{A}$, let X be a belief in $E_{\mathcal{A}}$. Then, M_X has a unique stationary distribution denoted $\sigma_X : X \to [0, 1]$, i.e. $\sigma_X \cdot M_X = \sigma_X$. Further, for all $y \in X$, $\sigma_{y,i} \xrightarrow[i \to +\infty]{} \sigma_X$.*

Proof sketch. We apply the fundamental theorem to M_X to get the statement. It suffices to show that M_X is ergodic. For all $x \in X$, by Lemma 3, there is an observation v_x leading from $\{x\}$ to X, i.e. $\Delta(\{x\}, v_x) = X$. As $\Delta(\{x\}, v_x^i)$ is increasing with i and $|\Delta(\{x\}, v_x^i)| \leq n$ for all i, we obtain $\Delta(\{x\}, v_x^{n+1}) = \Delta(X, v_x^{n+1})$. We can then obtain a word w_x with $\Delta(\{x\}, w_x) = \Delta(X, w_x) = X$. Now, by induction on the size of X, we can build a uniform word w such that $\Delta(\{x\}, w) = X$ for all $x \in X$. For all $x, y \in X$, we get $M_X^{|w|}(x, y) > 0$. ◀

4 Limit-sure Classifiability

We start by stating the definition of *limit-sure classification* more precisely:

▶ **Definition 5.** *Two HMMs $\mathcal{A}_1, \mathcal{A}_2$ are limit-sure classifiable iff there exists a computable function, also called a classifier, $f : \Sigma^* \to \{1, 2\}$ such that $P(\rho$ run of $\mathcal{A}_1$ of size $k \mid f(obs(\rho)) = 2) \to_{k \to \infty} 0$, and similarly for ρ run of $\mathcal{A}_2$.*

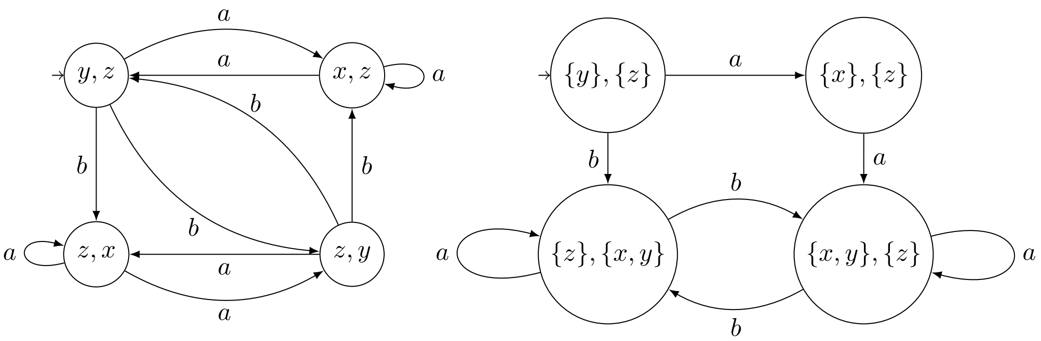

Figure 3 Twin automaton (on the left) and twin-belief automaton (on the right), for $\mathcal{A}_1, \mathcal{A}_2$ starting in states y and z.

(Notice we do not need $\bot$ as the classifier is allowed to give erroneous answers at first). Consider the *Maximum A Posteriori (MAP)* classifier [12, 10]: it answers 1 if $P^{\mathcal{A}_1}(u) > P^{\mathcal{A}_2}(u)$, and 2 otherwise. To do so, it just needs to record for every state of $\mathcal{A}_1$ (resp. every state of $\mathcal{A}_2$) the probability to observe u and finish in state s_1 (resp. s_2). Indeed, we may then compute confidence$(i, u) = \frac{P^{\mathcal{A}_i}(u)}{P^{\mathcal{A}_1}(u)+P^{\mathcal{A}_2}(u)}$, i.e. the probability that the decision i is correct after observing u. Notice that this confidence is not necessarily non-decreasing, and that the answer of a classifier can also switch from one answer to the other. In fact, we show in Proposition 16 in [1] that if $(\mathcal{A}_1, \mathcal{A}_2)$ is limit-sure classifiable, then the MAP classifier will be a limit-sure classifier. The main problem is to decide when limit-sure classification holds. In fact, this problem can be solved in PTIME. We remark that a variant of the problem was already shown to be in PTIME, namely distinguishability [6, 11]. While both problems coincide for HMMs, as explained in Section 4.4, our proof described in the rest of this section, crucially uses the notion of stationary distributions for HMMs developed in the previous section.

4.1 The Twin Automaton and the Twin Belief Automaton

Given HMMs $\mathcal{A}_1, \mathcal{A}_2$, we define their *twin automaton* $\mathcal{A} = (S = S_1 \times S_2, \Delta, s_0)$ as the product of the automata associated with $\mathcal{A}_1 \times \mathcal{A}_2$ by forgetting the probabilities. Recall that $\mathcal{A}_1$ has n states and $\mathcal{A}_2$ has m states. The transition relation is $\Delta = \{((s_1, s_2), a, (t_1, t_2)) \mid M_{\mathcal{A}_1}(s_1, a, t_1) > 0, M_{\mathcal{A}_2}(s_2, a, t_2) > 0\}$, with initial state $s_0 = (s_0^1, s_0^2)$. We call states of $\mathcal{A}$ *twin states*. In the following, we will often consider the belief automata $\mathcal{B}_\mathcal{A}, \mathcal{B}_{\mathcal{A}_1}, \mathcal{B}_{\mathcal{A}_2}$ associated with $\mathcal{A}, \mathcal{A}_1, \mathcal{A}_2$, obtained by the subset construction (see Section 3). States of $\mathcal{B}_\mathcal{A}$ will be called *twin beliefs*. Notice that although twin beliefs are formally sets of pairs of states in $2^{S_1 \times S_2}$, we can also present them as pairs of sets of states $2^{S_1} \times 2^{S_2}$ because if (s_1, s_2) and (s_1', s_2') are in the same twin belief, then we also have (s_1, s_2') and (s_1', s_2) in this twin belief. We will thus write the twin belief $X(u)$ associated with observation u as $X(u) = (X_1(u), X_2(u))$, with $X_1(u), X_2(u)$ the beliefs states of $\mathcal{B}_{\mathcal{A}_1}, \mathcal{B}_{\mathcal{A}_2}$ associated with u. Figure 3 presents an example with a twin automaton and the twin belief automaton for two copies of the HMM given in figure 1, one starting in state y and the other starting in state z.

▶ **Lemma 6** (Proposition 18 in [6]). *Let (X_1', X_2') be a reachable twin belief of $\mathcal{B}_\mathcal{A}$. Let $X_1 \subseteq X_1', X_2 \subseteq X_2'$. Let σ_1, σ_2 be two distributions over X_1, X_2 with $(\mathcal{A}_1, \sigma_1) \equiv (\mathcal{A}_2, \sigma_2)$. Then one cannot classify between $\mathcal{A}_1, \mathcal{A}_2$.*

4.2 Characterization for classifiability

Our goal is to use the result of Section 3 to obtain stationary distributions in $\mathcal{A}_1, \mathcal{A}_2$, and classify between them by comparing the stochastic language wrt these stationary distributions using probabilistic equivalence (see Section 2.1). In order to do this, we first need to compare the same information in both HMMs. The idea is to consider twin beliefs from each HMM: we will enrich $\mathcal{A}_1$ with the beliefs of $\mathcal{A}_2$, and vice versa. Let $\mathcal{A}'_1$ be the HMM where the state space is $S_1 \times 2^{S_2}$, and the transition matrix is $M_{\mathcal{A}'_1}((x,Y),a,(x',Y')) = M_{\mathcal{A}_1}(x,a,x')$ if $Y' = \{y' \mid (y,a,y'), y \in Y\}$, and 0 otherwise, for all x, Y, a, x', Y'. We define similarly $\mathcal{A}'_2$ with set of states $S_2 \times 2^{S_1}$. It is easy to see that for all observation w, the belief state $B_{\mathcal{A}'_1}(w) = \{(x_1, B_{\mathcal{A}_2}(w)) \mid x_1 \in B_{\mathcal{A}_1}(w)\}$, is isomorphic to the twin belief $(B_{\mathcal{A}_1}(w), B_{\mathcal{A}_2}(w))$, isomorphic to $B_{\mathcal{A}'_2}(w)$, and we will abuse notation and represent beliefs of $\mathcal{A}'_1$ and $\mathcal{A}'_2$ as twin belief (X_1, X_2), where X_1 or X_2 can be empty.

What we are interested in is what happens after a BSCC of $\mathcal{A}$ is reached. We thus consider twin beliefs reachable from some (x_1, x_2) in the BSCC of $\mathcal{A}$. The set of twin beliefs reachable in $\mathcal{A}'_1$ and in $\mathcal{A}'_2$ from $(\{x_1\}, \{x_2\})$ are almost the same, except for twin beliefs of the form $(X_1, \emptyset)$ which cannot be reached in $\mathcal{A}'_2$, and of the form $(\emptyset, X_2)$ which cannot be reached in $\mathcal{A}'_1$.

▶ **Definition 7.** *We say that a twin belief (X_1, X_2) is* oblivious *if the languages of $\mathcal{B}_{\mathcal{A}_1}$ from X_1 and of $\mathcal{B}_{\mathcal{A}_2}$ from X_2 are the same.*

By definition, if (X_1, X_2) is not oblivious, there are words differentiating X_1 and X_2.

Now, assume that $X = (X_1, X_2)$ is oblivious. The twin beliefs reachable from (X_1, X_2) are the same in $\mathcal{A}'_1$ and $\mathcal{A}'_2$. To potentially differentiate them, we need to consider their long term statistics. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the belief automata associated with $\mathcal{A}'_1$ and $\mathcal{A}'_2$. Let $E_{\mathcal{A}}$ be the union of BSCCs of twin beliefs accessible from twin states in the BSCCs of twin states, as in lemma 3. Let $X \in E_{\mathcal{A}}$. In this case, we say that X is in the BSCCs of twin beliefs. We define $\sigma^1_X : X_1 \to [0,1]$ the *stationary distribution* in $\mathcal{A}'_1$ around the twin belief X (formally, σ^1_X is defined on (x, X_2) for all $x \in X_1$, and we omit the second component X_2 because it is constant). In the same way, we define $\sigma^2_X : X_2 \to [0,1]$ for the second component X_2 around the twin belief X. We can then look for words differentiating $\mathcal{A}_1, \mathcal{A}_2$, i.e. with different probabilities from σ^1_X and from σ^2_X. We can now state our characterization:

▶ **Theorem 8.** *The following are equivalent:*

1. *One cannot limit-surely classify between $\mathcal{A}_1, \mathcal{A}_2$,*
2. *There exists an oblivious $X \in E_{\mathcal{A}}$ in a BSCC of twin beliefs such that $(\mathcal{A}_1, \sigma^1_X) \equiv (\mathcal{A}_2, \sigma^2_X)$,*
3. *There exists a BSCC D of $\mathcal{A}$ and $X_1 \subseteq S_1, X_2 \subseteq S_2$, and $y_1 \in X_1$, $y_2 \in X_2$, such that $(y_1, x_2) \in D$ for all $x_2 \in X_2$ and $(x_1, y_2) \in D$ for all $x_1 \in X_1$, and two distributions σ^1 over X_1 and σ^2 over X_2 such that $(\mathcal{A}_1, \sigma^1) \equiv (\mathcal{A}_2, \sigma^2)$.*

The second condition is sufficient to show that MAP is a limit-sure classifier (see Proposition 16 in [1]). However, checking condition 2 explicitly is not algorithmically efficient, as the belief automaton can have exponentially many states. Instead, to obtain a PTIME algorithm to check limit-sure classifiability, we will use the third condition. For comparison, in [6], a variant of the equivalence between (1) and (3) is shown, without using the stationary distributions σ^1_X, σ^2_X of (2).

For the proof, we note that the case of 2 implies 3 is easy. For the remaining two directions, i.e. 1 implies 2 and 3 implies 1, proofs are technical, and can be found in the long version [1]. For 1 implies 2, we prove that negation of 2 implies that the MAP classifier (defined in beginning of Section 4) is limit-sure, implying negation of 1. Intuitively, negation of 2 means that every pair of reachable beliefs have a distinguishing word. It then suffices to consider statistics on these finite number of distinguishing words to know the originating HMM with arbitrarily high probability. For 3 implies 1, we show that any twin belief (H_1, H_2) reached from (y_1, y_2) in $E_{\mathcal{A}}$ must be oblivious because of the probabilistic equivalence. We show this implies $(\mathcal{A}_1, \sigma^1_{H_1,H_2})$ and $(\mathcal{A}_2, \sigma^2_{H_1,H_2})$ are equivalent and conclude using Lemma 6.

4.3 A PTIME Algorithm

Theorem 8 gives us a characterization for the existence of a limit-sure classifier. The third condition is particularly interesting, because it does not require computing beliefs. Using this, we can build an efficient algorithm, similar to [6], to test in PTIME whether there exists a limit-sure classifier between $\mathcal{A}_1, \mathcal{A}_2$.

Our Algorithm 1, presented below, uses linear programming: we let $v_1, \dots, v_\ell$ be the basis of $Eq(\mathcal{A}_1, \mathcal{A}_2)$ (see Section 2.1). There exist two distributions σ^1, σ^2 over X_1, X_2 with $(\mathcal{A}_1, \sigma^1) \equiv (\mathcal{A}_2, \sigma^2)$ iff the linear system of equations (for all $j \leq \ell$, $(\sigma^1 \quad \sigma^2) \cdot v_j = 0$) has a solution (with σ^1, σ^2 as variables), which can be solved in Polynomial time.

Algorithm 1 Limit-sure Classifiability.

```
Compute $D_1, \dots, D_k$ the BSCCs of the twin automaton $\mathcal{A}$.
for i=1..k do
    for $(y_1, y_2) \in D_i$ do
        Let $X_1 = \{x_1 \mid (x_1, y_2) \in D_i\}$, $X_2 = \{x_2 \mid (y_1, x_2) \in D_i\}$.
        if there exist two distributions $\sigma^1, \sigma^2$ over $X_1, X_2$ with $\sigma^1(y_1) > 0$ and $\sigma^2(y_2) > 0$
            with $(\mathcal{A}_1, \sigma^1) \equiv (\mathcal{A}_2, \sigma^2)$ then
            return not classifiable
return classifiable
```

The correctness of the algorithm is immediate from Theorem 8, as it checks explicitly for the third condition to hold, in which case it returns not classifiable. If the third condition is false for every BSCC D, then it returns classifiable.

4.4 Comparison with Distinguishability between HMMs [11]

We complete this section, by comparing our results with a related result on HMMs. In [11], the problem of distinguishability between labeled Markov Chains has been considered. First, labeled Markov Chains are just another name for HMMs. The idea behind distinguishability is similar to the idea behind classifiability. Still, there are some technical differences: distinguishability asks that for all $\varepsilon > 0$, there exists a $(1-\varepsilon)$-*classifier*, that is a classifier $f : \Sigma^* \to \{\bot, 1, 2\}$, such that if the classifier answers $f(u) = 1$, then there is probability at least $(1-\varepsilon)$ that the observation comes from a run from $\mathcal{A}_1$, and similarly for $f(u) = 2$. To compare, limit-sure classifiers need to be uniform over ε (see the next section).

The authors of [11] show that this notion can be checked in PTIME, by indirectly using the result of [6] stating that one can check in PTIME whether the total variation distance between two HMMs is 1. More precisely, the total variation distance is defined as:

▶ **Definition 9.** *The* total variation distance *between two HMMs $\mathcal{A}_1$ and $\mathcal{A}_2$ is given by*

$$d(\mathcal{A}_1, \mathcal{A}_2) = \sup_{E \subset \Sigma^\omega} |P_{\mathcal{A}_1}(E) - P_{\mathcal{A}_2}(E)|.$$

This supremum has been shown to be a maximum [6]. It is not too hard to show that limit-sure classification coincides with these notions as well for HMMs:

▶ **Theorem 10.** *The following are equivalent:*

1. *There exists a limit-sure classifier for* $\mathcal{A}_1, \mathcal{A}_2$*,*
2. *For all* $\varepsilon > 0$*, there exists a* $(1-\varepsilon)$-classifier *for* $\mathcal{A}_1, \mathcal{A}_2$*,*
3. $d(\mathcal{A}_1, \mathcal{A}_2) = 1$*.*

The proofs to obtain the PTIME algorithms are quite different though: we use stationary distributions in HMMs while [6] focuses on separating events. Some intermediate results are however related: our Proposition 18 in [1] is to be compared with Proposition 19 b) of [6]: Our statement is stronger as the equivalence is true from *all* pairs of states with the same (non stochastic) language - and in particular from $(i_1, j_1) = (y_1, y_2)$ (cf Proposition 17 in [1]). Also, the proof of Proposition 18 in [1] is simple, using strict convexity focusing on *one* finite separating word, while in [6], the existence of a maximal separating events (sets of infinite words) is used crucially in the proof of Proposition 19 b).

Surprisingly, our resulting algorithm is very similar to the one in [6], whereas we use very different methods. Still, we can restrict the search to *distributions* in a BSCC of twin states, while [6] considers *subdistributions* on the whole state space of twin states. This allows us to optimize the number of variables in the Linear Program.

5 Attack-classification

While limit-sure classification allows for some misclassification, i.e. error in classification, it requires that every execution of the HMMs is classifiable. From a security perspective, if one wants to make sure that two systems cannot be distinguished from each other, then the question changes slightly: from the point of view of an attacker who could exploit the knowledge of which model the system is following, it need not classify every single execution. It only needs to find one execution for which it can decide. This gives rise to what we call *attack-classification*, which amounts to providing the attacker with a reset action she can play when she believes the execution cannot be classified. Then, a new (possibly the same) HMM is taken at random and an execution of this new HMM is observed by the attacker. For instance, it is not possible to limit-surely classify between HMM $\mathcal{A}_3$ and HMM $\mathcal{A}_4$ on Figure 4, because executions starting with a b cannot be classified. On the other hand, an attacker can wait for an execution of the system starting with an a, for which he is sure the HMM is $\mathcal{A}_3$. If it starts with a b, then the attacker just forgets this execution and wait for a new execution of the system (the "reset" operation).

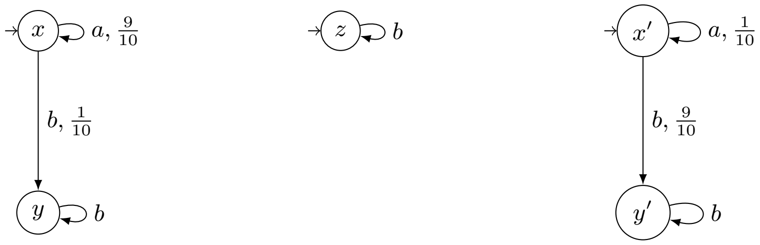

Figure 4 HMMs $\mathcal{A}_3, \mathcal{A}_4$ and $\mathcal{A}_5$ (left to right). One cannot classify betweeen $\mathcal{A}_3, \mathcal{A}_4$, but they can be attack-classified. On the other hand, one cannot attack-classify between $\mathcal{A}_3, \mathcal{A}_5$.

We start by considering *limit-sure attack-classifiers*, namely, we require that there exists a *reset-strategy*, which with probability 1, resets only finitely many times, and a limit-sure classifier for the observation after the last reset. We also consider what happens if instead of limit-sure classifier, we ask for the existence of a family of $(1-\varepsilon)$-classifiers after the last reset, one for each ε. The difference is that the reset action can take into account the ε in the latter, but not in the former. While both notions coincide for the classifiers defined in the previous section, we show now that they do not coincide for attack-classification.

Figure 4 illustrates the difference between these two notions, considering $\mathcal{A}_3$ and $\mathcal{A}_5$. First, for all $\varepsilon > 0$, there exists an $(1-\varepsilon)$-attack-classifier: given an ε, the reset strategy resets if the first letter b happens within the first $k_\varepsilon = log(\frac{1}{9\varepsilon})$ steps. That is, the reset strategy is $\tau(a^*) = \perp$, $\tau(a^{k_\varepsilon}w) = \perp$ and $\tau(a^\ell b) = reset$ for $\ell < k_\varepsilon$. For observation $a^{k_\varepsilon}w$, the classifier claims that the HMM is $\mathcal{A}_3$, which is true with probability at least $(1-\varepsilon)$. However, this reset strategy is not compatible with limit-sure classifier (and, in fact, no reset strategy is), because it is not uniform wrt all ε: once a b has been produced, no more information can be gathered. On the other hand, limit-sure attack-classified implies the existence of $(1-\varepsilon)$-attack-classifiers for all ε. Thus the former notion of limit-sure attack-classifier is strictly contained in the latter. More importantly, we show that deciding the former is PSPACE-complete, while the latter turns out to be undecidable.

5.1 Limit-sure attack-classifiability is PSPACE-complete

Let us first formalize the definition of attack-classification.

▶ **Definition 11.** *We say two HMMs $\mathcal{A}_1, \mathcal{A}_2$ are* limit-sure attack-classifiable *if: there exists*

1. reset strategy $\tau : \Sigma^* \to \{\perp, reset\}$ *telling when to reset, and which eventually stops resetting, with probability 1 on the reset runs, and*
2. *limit-sure classifier for u, where $u \in \Sigma^*$ denotes the suffix of observations since last reset.*

In the following, we show an algorithmic characterization for this concept. Intuitively, there needs to exist one execution of one HMM (say $\mathcal{A}_1$), such that no matter the execution of the other HMM with the same observation, we can eventually classify between these two executions. We will thus consider $\mathcal{A}'_1$ and $\mathcal{A}'_2$, the HMMs $\mathcal{A}_1$ and $\mathcal{A}_2$ enriched with the beliefs of the other HMM.

First, we define classifiable twin states in the BSCC of twin states: $(x_1, x_2) \in \mathcal{A}$ is classifiable iff for (X_1, X_2) in the unique BSCC of twin beliefs, either (X_1, X_2) is non oblivious or (X_1, X_2) is oblivious and $(\mathcal{A}_1, \sigma^1_{X_1,X_2}) \not\equiv (\mathcal{A}_2, \sigma^2_{X_1,X_2})$, for $(\sigma^1_{X_1,X_2}, \sigma^2_{X_1,X_2})$ the stationary distributions built for (X_1, X_2). Notice that it does not depends upon the choice of (X_1, X_2). For a belief state X_2 of $\mathcal{A}_2$, we say that $(x_1, X_2) \in \mathcal{A}'_1$ is classifiable if (x_1, x_2) is classifiable for all $x_2 \in X_2$ (in particular, every (x_1, x_2) is in a BSCC of twin states). In particular, $(x_1, \emptyset)$ is classifiable. We define $(x_2, X_1) \in \mathcal{A}'_2$ similarly.

▶ **Proposition 12.** $(\mathcal{A}_1, \mathcal{A}_2)$ *is limit-sure attack-classifiable iff there exists a classifiable* $(x_1, X_2) \in \mathcal{A}'_1$*, or a classifiable* $(x_2, X_1) \in \mathcal{A}'_2$*.*

In case there are more than two HMMs, we follow the state s of one HMM and the belief of every other HMMs along the observation, and we need to check classifiability between (s, t) for every t in the belief of any of the other HMMs. Using this characterization, we obtain:

▶ **Theorem 13.** *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be two HMMs. It is PSPACE-complete to check whether $(\mathcal{A}_1, \mathcal{A}_2)$ are limit-sure attack-classifiable.*

5.2 Existence of $(1 - \varepsilon)$ attack-classifiers for all ε is undecidable

We now turn to the other notion. Let $\varepsilon > 0$. An $(1 - \varepsilon)$ attack-classifier for two HMMs $\mathcal{A}_1, \mathcal{A}_2$ is given by:

1. A reset strategy $\tau : \Sigma^* \to \{\bot, reset\}$ telling when to reset, and which eventually stops resetting, with probability 1 on the reset runs, and
2. a $(1 - \varepsilon)$-classifier for u, where $u \in \Sigma^*$ denotes the suffix of the observations since the last reset.

We next show that this notion, which we showed to be weaker than limit-sure attack-classifiability on Fig 4, is also computationally much harder, in fact, it is undecidable.

▶ **Theorem 14.** *It is undecidable to know whether for all ε, there exists an $(1 - \varepsilon)$ attack-classifier between 2 HMMs.*

Intuitively, we reduce from the problem of whether a PFA $\mathcal{B}$, that accepts all words with probability in $(0, 1)$, is 0 and 1 isolated, that is, there is no sequence of words $(w_i)_{i\in\mathbb{N}}$ such that $lim_{n\to\infty} P^{\mathcal{B}}(w_i) = 0$ or $= 1$. This problem is undecidable [8]. The idea is to transform the PFA into an HMM which performs the actions of the PFA uniformly at random. We check whether we can attack classify this HMM with an HMM which accepts all words of size k with probability $1/2^k$. This is possible if 0 is not isolated or if 1 is not isolated.

6 Conclusion

In this paper, we tackled the notion of limit-sure classifiability between HMMs, which is a general notion in studying how to uncover hidden information in partially observable systems. The class of classifiers we consider are quite powerful, as they can use statistics on the observations in order to take their decision. To obtain our results, summarized in the table below we developed a robust theory of stationary distributions for HMMs.

While limit-sure classifiability is stronger and more complex than almost-sure classifiability, checking for it is in a lower complexity class: PTIME instead of PSPACE-complete. This result shines some new light on total variation metric for stochastic systems, recovering with different techniques the PTIME result from [6]. We also considered attack-classifiability, where the attacker needs to classify at least one observation rather than every execution. In this setting, there is a difference between limit-sure classifier and the existence of $(1 - \varepsilon)$-classifiers for each ε. Limit-sure attack-classifiability is decidable (PSPACE-complete), whereas the existence of $(1 - \varepsilon)$-classifiers for all ε is undecidable.

	limit-sure classifiability	limit-sure attack-classifiability	$\forall\varepsilon, (1 - \varepsilon)$ attack-classifiability
Complexity	PTIME	PSPACE-complete	Undecidable

References

1 S. Akshay, Hugo Bazille, Eric Fabre, and Blaise Genest. Classification among Hidden Markov Models, 2019. URL: `http://perso.crans.org/genest/ABFG19.pdf`.

2 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, Cambridge, MA, 2008.

3 Vijay Balasubramanian. Equivalence and Reduction of Hidden Markov Models, 1993.

4 Nathalie Bertrand, Serge Haddad, and Engel Lefaucheux. Foundation of diagnosis and predictability in probabilistic systems. In *34th International Conference on Foundation of Software Technology and Theoretical Computer Science*, volume 29 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 417–429. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2014.

5 Nathalie Bertrand, Serge Haddad, and Engel Lefaucheux. Accurate approximate diagnosability of stochastic systems. In *Language and automata theory and applications*, volume 9618 of *Lecture Notes in Comput. Sci.*, pages 549–561. Springer, 2016. doi:10.1007/978-3-319-30000-9_42.

6 Taolue Chen and Stefan Kiefer. On the total variation distance of labelled Markov chains. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages Article No. 33, 10. ACM, New York, 2014.

7 Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Equivalence of labeled Markov chains. *Internat. J. Found. Comput. Sci.*, 19(3):549–563, 2008. doi:10.1142/S0129054108005814.

8 Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: decidable and undecidable problems. In *Automata, languages and programming. Part II*, volume 6199 of *Lecture Notes in Comput. Sci.*, pages 527–538. Springer, Berlin, 2010. doi:10.1007/978-3-642-14162-1_44.

9 John G. Kemeny and J. Laurie Snell. *Finite Markov chains.* The University Series in Undergraduate Mathematics. D. Van Nostrand Co., Inc., Princeton, N.J.-Toronto-London-New York, 1960.

10 Christoforos Keroglou and Christoforos N. Hadjicostis. Probabilistic system opacity in discrete event systems. *Discrete Event Dyn. Syst.*, 28(2):289–314, 2018. doi:10.1007/s10626-017-0263-8.

11 Stefan Kiefer and A. Prasad Sistla. Distinguishing hidden Markov chains. In *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)*, page 10. ACM, New York, 2016.

12 Daniel Ramage. Hidden Markov Models Fundamentals, CS229, lecture notes, 2007.

13 Anooshiravan Saboori and Christoforos N. Hadjicostis. Probabilistic current-state opacity is undecidable. In *Proceedings of the 19th Intl. Symposium on Mathematical Theory of Networks and Systems*, pages 1–10, 2010.

14 Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. Contr. Sys. Techn.*, 4(2):105–124, 1996. doi:10.1109/87.486338.

15 David Thorsley and Demosthenis Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Trans. Automat. Control*, 50(4):476–492, 2005. doi:10.1109/TAC.2005.844722.

16 Wen-Guey Tzeng. The Equivalence and Learning of Probabilistic Automata (Extended Abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989, IEEE FOCS'89*, pages 268–273, 1989. doi:10.1109/SFCS.1989.63489.