Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

rsta.royalsocietypublishing.org

Research

Cite this article: Chattopadhyay I, Lipson H. 2013 Abductive learning of quantized stochastic processes with probabilistic finite automata. Phil Trans R Soc A 371: 20110543. http://dx.doi.org/10.1098/rsta.2011.0543

One contribution of 17 to a Discussion Meeting Issue 'Signal processing and inference for the physical sciences'.

Subject Areas:
applied mathematics

Keywords:
machine learning, probabilistic finite automata, stochastic processes, symbolic dynamics

Author for correspondence:
Hod Lipson
e-mail: hod.lipson@cornell.edu

Royal Society Publishing
Informing the science of the future

Abductive learning of quantized stochastic processes with probabilistic finite automata

Ishanu Chattopadhyay and Hod Lipson

Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA

We present an unsupervised learning algorithm (GenESeSS) to infer the causal structure of quantized stochastic processes, defined as stochastic dynamical systems evolving over discrete time, and producing quantized observations. Assuming ergodicity and stationarity, GenESeSS infers probabilistic finite state automata models from a sufficiently long observed trace. Our approach is abductive; attempting to infer a simple hypothesis, consistent with observations and modelling framework that essentially fixes the hypothesis class. The probabilistic automata we infer have no initial and terminal states, have no structural restrictions and are shown to be probably approximately correct-learnable. Additionally, we establish rigorous performance guarantees and data requirements, and show that GenESeSS correctly infers long-range dependencies. Modelling and prediction examples on simulated and real data establish relevance to automated inference of causal stochastic structures underlying complex physical phenomena.

1. Introduction and motivation

Automated model inference is a topic of wide interest, and reported techniques range from query-based concept learning in artificial intelligence [1,2], to classical system identification [3], to more recent symbolic regression based reverse-engineering of nonlinear dynamics [4,5]. In this paper, we wish to infer the dynamical structure of quantized stochastic processes (QSPs): stochastic dynamical systems evolving over discrete time, and producing quantized time series (figure 1). Such systems naturally arise via quantization of successive increments of discrete time continuous-valued stochastic processes.

© 2012 The Author(s) Published by the Royal Society. All rights reserved.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

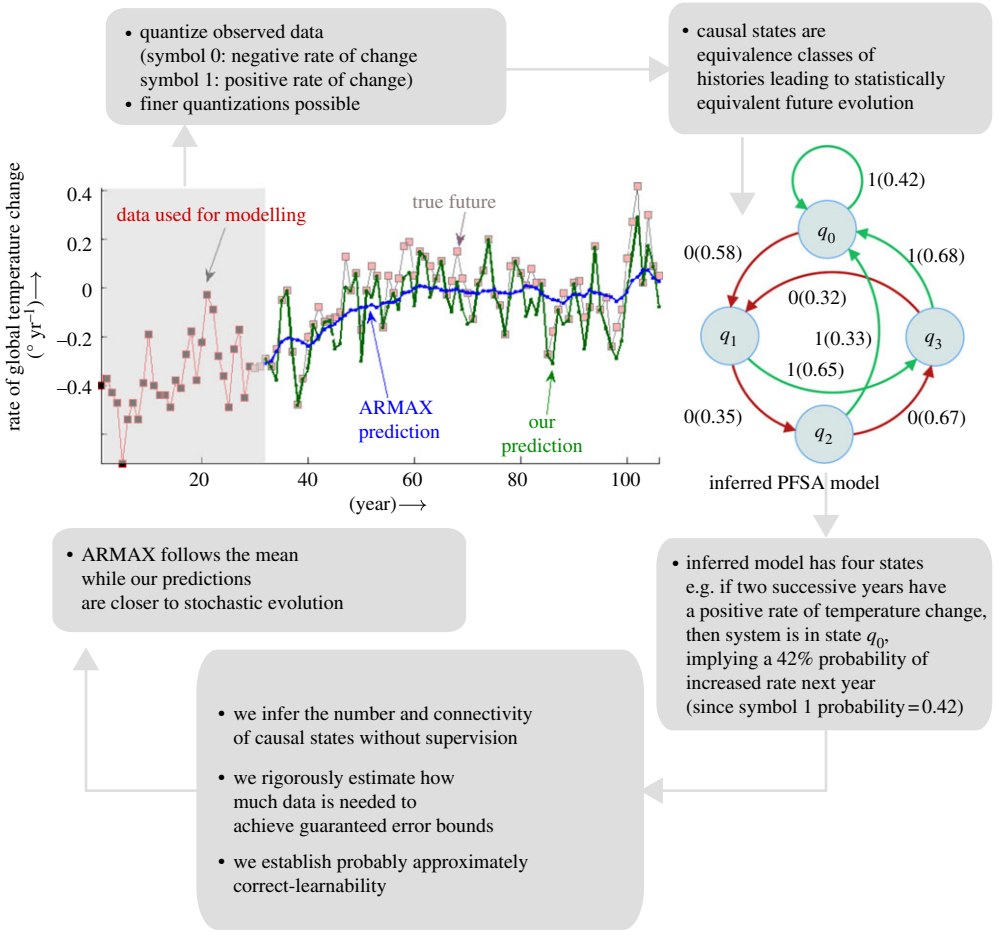


Figure 1. Inference in physical systems: learning causal structure as a probabilistic finite state automaton (PFSA) from quantized time series of annual global temperature changes from 1880 to 1986. Inferred PFSA, and comparative predictions against an auto-regressive moving-average (ARMAX) model are shown (predicting for 1 year in the future) overlaid with the true observations. (Online version in colour.)

A simple example is the standard random walk on the line (figure 2). Constant positive and negative increments are represented symbolically as σ_R and σ_L, respectively, leading to a stochastic dynamical system that generates strings over the alphabet $\{\sigma_L, \sigma_R\}$, with each new symbol being generated independently with a probability of 0.5. The dynamical characteristics of this process are captured by the probabilistic automaton shown in figure 2*b*, consisting a single state q_1, and self-loops labelled with symbols from Σ, which represent probabilistic transitions. We wish to infer such causal structures, formally known as probabilistic finite state automaton (PFSA), from a sufficiently long sequence of observed symbols, with no *a priori* knowledge of the number, connectivities and the transition rules of the hidden states.

(a) Relevance to signal processing and inference in physical systems

We use symbolic representation of data, and our inferred models are abstract logical machines (PFSA). We quantize relative changes between our successive physical observations to map the gradient of continuous time series to symbolic sequences (with each symbol representing a specific increment range). Additionally, we want our inferred models to be generative, i.e. predict the distribution of the future symbols from knowledge of the recent past. What do we gain from

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

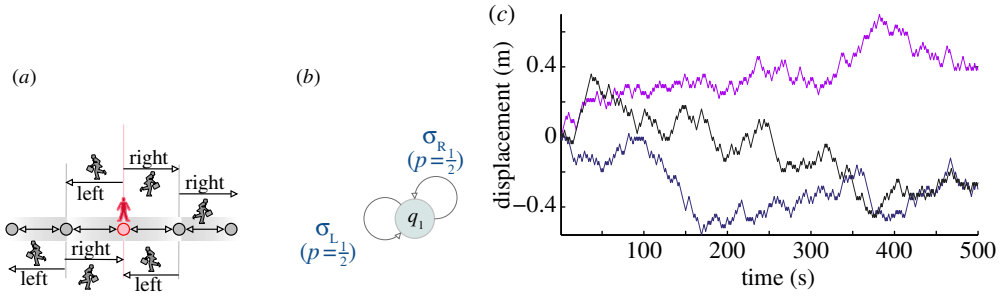


Figure 2. Simple example of a QSP: a random walk on the line (*a*). The 1-state probabilistic automaton shown in (*b*) captures the statistical characteristics of this process. Our objective is to learn such a model from a single long observed trace, like any one of the three shown in (*c*). (Online version in colour.)

such a symbolic approach, particularly since we incur some information loss in quantization? The main advantages are unsupervised inference, computational efficiency, deeper insight into the hidden causal structures, ability to deduce rigorous performance guarantees and the ability to better predict stochastic evolution dictated by many hidden variables interacting in a structured fashion. Systems with such stochastic dynamics and complex causal structures are of emerging interest, e.g. flow of wealth in the stock market, geophysical phenomena, ecology of evolving ecosystems, genetic regulatory circuits and even social interaction dynamics.

Our scheme is illustrated in figure 1, where we model global annual temperature changes from 1880 to 1986 [6]. GenESeSS infers a 4-state PFSA; each state specifies the probabilities of positive (symbol 1) and negative (symbol 0) rate of change in the coming year. Here, we use a binary alphabet, but finer quantizations are illustrated in §4*b*. Global temperature changes result from the complex interaction of many variables, and a first principle model is not deducible from this data alone; but we can infer causal states in the data as equivalence classes of history fragments yielding statistically similar futures.

We use the following intuition: if the recent pattern of changes was encountered in the past, at least approximately, and then often led to a positive temperature change in the immediate future, then we expect to see a positive change in the coming year as well. Causal states simply formalize the different classes of patterns that need to be considered, and are inferred to specify which patterns are equivalent.

This allows us to make quantitative predictions. For example, state q_0 is the equivalence class of histories with last two consecutive years of increasing temperature change (two 1 s), and our inferred model indicates that this induces a 42 per cent probability of a positive rate of change next year (that such a pattern indeed leads uniquely to a causal state is inferred from data, and not manually imposed). Additionally, the maximum deviation between the inferred and the true hidden model is limited by a user-selectable bound. For comparison, we also learn a standard auto-regressive moving-average (ARMAX) model [7], which actually predicts the mean quite well. However, we capture stochastic changes better, and gain deeper insight into the causal structure in terms of the number of equivalence classes of histories, their inter-relationships and model memory.

(b) Abduction versus induction

Our modelling framework assumes that, at any given instant, a QSP is in a *unique hidden causal state*, which has a well-defined probability of generating the next quantized observation. This fixes the *rule* (i.e. the hypothesis class) by which sequential observations are generated, and we seek the correct hypothesis, i.e. the automaton that explains the observed trace. Inferring the hypothesis, given the observations and the generating rule is *abduction* (as opposed to induction, which infers generating rules from a knowledge of the hypothesis, and the observation set) [8].

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

(c) Contribution and contrast with reported literature

Learning from symbolic data is essentially grammatical inference; learning an unknown formal language from a presentation of a finite set of example sentences [9–11]. In our case, every subsequence of the observed trace is an example sentence [12]. Inference of probabilistic automata is well studied in the context of pattern recognition. However, learning dynamical systems with PFSA has received less attention, and we point out the following issues.

— *Initial and terminal conditions.* Dynamical systems evolve over time, and are often termination-free. Thus, inferred machines should lack terminal states. Reported algorithms [13–15] often learn probabilistic automata with initial and final states, and with special termination symbols.
— *Structural restrictions.* Reported techniques make restrictive structural assumptions, and pre-specify upper bounds on inferred model size. For example, [13,15,16] infer only probabilistic suffix automata, those in [12,17] require synchronizability, i.e. the property by which a bounded history is sufficient to uniquely determine the current state, and [13] restrict models to be acyclic and aperiodic.
— *Restriction to short memory processes.* Often memory bounds are pre-specified as bounds on the order of the underlying Markov chain [16], or synchronizability [12]; thus only learning processes with *short range dependencies*. A time series possesses *long-range dependencies* (LRDs), if it has correlations persisting over all time scales [18]. In such processes, the auto-correlation function follows a power law, as opposed to an exponential decay. Such processes are of emerging interest, e.g. internet traffic [19], financial systems [20,21] and biological systems [22,23]. We can learn LRD processes (figure 3).
— *Inconsistent definition of probability space.* Reported approaches often attempt to define a probability distribution on $\Sigma^\star$, i.e. the set of all finite but unbounded words over an alphabet Σ, and then additionally *define* the probability of the null-word λ to be 1. This is inconsistent, since the latter condition would demand that the probability of every other finite string in $\Sigma^\star$ be 0. Some authors [14] use λ for string termination, which is in line with the grammatical picture where the empty word is used to erase a non-terminal [26]. However, this is inconsistent with a dynamical systems model. We address this via rigorous construction of a σ-algebra on strictly infinite strings.

Our algorithmic procedure (see §3*a*) has some commonalities with [27,28]. However, Gavaldà *et al*. uses initial states (superfluous for stationary ergodic processes), special termination symbols and requires a pre-specified bound on the model size. Additionally, key technical differences (discussed in §3*a*) make our models significantly more compact.

To summarize our contribution, (i) we formalize PFSAs in the context of QSPs, (ii) remove inconsistencies with the definition of the probability of the null-word via a σ-algebra on strictly infinite strings, and (iii) show that PFSAs arise naturally via an equivalence relation on infinite strings. Also, (iv) we characterize the class of QSPs with finite probabilistic generators, establish probably approximately correct (PAC)-learnability [29] and show that (v) algorithm `GenESeSS` learns PFSA models with no *a priori* restrictions on the structure, size and memory.

The rest of the paper is organized as follows: §2 develops new theory, which is used in §3 to design `GenESeSS` and establish complexity results within the PAC criterion. Section 4 validates our development using synthetic and experimental data. The paper is summarized and concluded in §5.

2. Formalization of probabilistic generators for stochastic dynamical systems

Throughout this paper, Σ denotes a finite alphabet of symbols. The set of all finite but possibly unbounded strings on Σ is denoted by $\Sigma^\star$, the Kleene $\star$ operation [26]. The set of finite strings over Σ form a concatenative monoid, with the empty word λ as identity. Concatenation of two

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

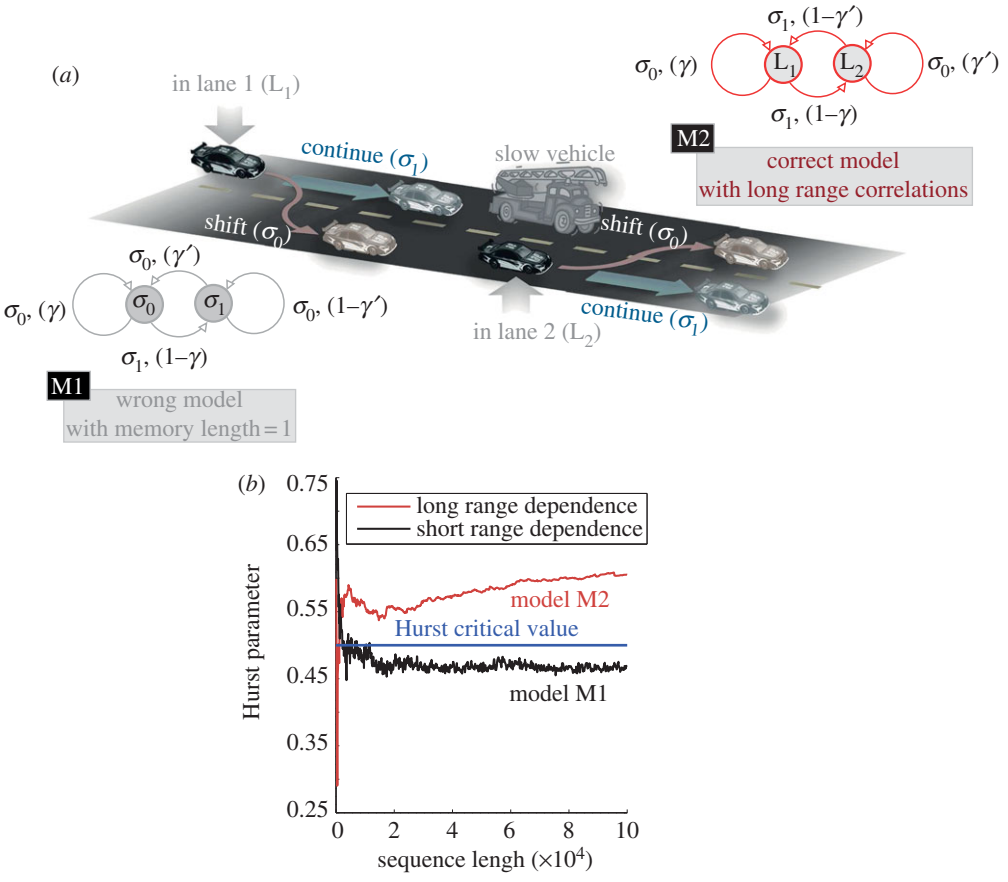


Figure 3. Process with LRDs. (*a*) Consider the dynamics of lane switching for a car on the highway, which may shift lanes (symbol σ_0) or continue in the current lane (symbol σ_1), and these symbol probabilities depend on the lane that the car is in. The correct model for this rule is M2 (top), which is non-synchronizable, and no length of history will determine exactly the current state. The similar, but incorrect, model M1 (with transitions switched) is a PFSA with 1-step memory. The difference in dynamics is illustrated by the computation of the Hurst exponent for symbol streams generated by the two models (*b*), with σ_0 and σ_1 represented as 1 and −1. M2 has a Hurst exponent greater than 0.5 [24,25], indicating LRD behaviour. GenESeSS correctly identifies M2 from a sequence of driver decisions; while most reported algorithms identify models similar to M1. (Online version in colour.)

strings $x, y \in \Sigma^\star$ is written as xy. Thus, $xy = x\lambda y = xy\lambda = \lambda xy$. The set of strictly infinite strings on Σ is denoted as Σ^ω, where ω denotes the first transfinite cardinal. For a string $x \in \Sigma^\star$, $|x|$ denotes the length of x and for a set A, $|A|$ denotes its cardinality.

(a) Quantized stochastic processes

Definition 2.1 (QSP). A QSP $\mathcal{H}$ is a discrete time Σ-valued strictly stationary, ergodic stochastic process, i.e.

$$\mathcal{H} = \{X_t : X_t \text{ is a } \Sigma\text{-valued random variable for } t \in \mathbb{N} \cup \{0\}\}.$$

A stochastic process is ergodic if moments can be calculated from a single, sufficiently long realization, and strictly stationary if moments are not functions of time.

We next formalize the connection of QSPs to PFSA generators.

Definition 2.2 (σ-algebra on infinite strings). For the set of infinite strings on Σ, we define $\mathbb{B}$ to be the smallest σ-algebra generated by $\{x\Sigma^\omega : x \in \Sigma^\star\}$.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

Lemma 2.3. *Any QSP induces a probability space* $(\Sigma^\omega, \mathbb{B}, \mu)$.

Proof. Using stationarity of QSP $\mathcal{H}$, we can construct a probability measure $\mu : \mathbb{B} \to [0,1]$ by defining for any sequence $x \in \Sigma^\star \setminus \{\lambda\}$, and a sufficiently large number of realizations N_R of $\mathcal{H}$, with fixed initial conditions:

$$\mu(x\Sigma^\omega) = \lim_{N_R \to \infty} \frac{\text{(number of initial occurrences of } x)}{\text{(number of initial occurrences of all sequences of length } |x|)}$$

and extending the measure to elements of $\mathbb{B} \setminus B$ via at most countable sums. Note that $\mu(\Sigma^\omega) = \sum_{x \in \Sigma^\star} \mu(x\Sigma^\omega) = 1$, and for the null word $\mu(\lambda\Sigma^\omega) = \mu(\Sigma^\omega) = 1$. ■

For notational brevity, we denote $\mu(x\Sigma^\omega)$ as $\Pr(x)$.

Classically, states are induced via the Nerode equivalence, which defines two strings to be equivalent if and only if any finite extension of the strings is either both accepted or both rejected [26] by the language under consideration. We use a probabilistic extension [30].

Definition 2.4 (probabilistic Nerode relation). $(\Sigma^\omega, \mathbb{B}, \mu)$ induces an equivalence relation $\sim_N$ on the set of finite strings $\Sigma^\star$ as

$$\begin{aligned} &\forall x, y \in \Sigma^\star, x \sim_N y \\ &\quad \iff \forall z \in \Sigma^\star \left((\Pr(xz) = \Pr(yz) = 0) \vee \left| \frac{\Pr(xz)}{\Pr(x)} - \frac{\Pr(yz)}{\Pr(y)} \right| = 0 \right). \end{aligned} \tag{2.1}$$

For $x \in \Sigma^\star$, the equivalence class of x is denoted as $[x]$.

It follows that $\sim_N$ is right invariant, i.e.

$$x \sim_N y \Rightarrow \forall z \in \Sigma^\star, \quad xz \sim_N yz. \tag{2.2}$$

A right-invariant equivalence on $\Sigma^\star$ always induces an automaton structure.

Definition 2.5 (initial-marked PFSA). An initial-marked PFSA is a 5-tuple $(Q, \Sigma, \delta, \tilde{\pi}, q_0)$, where Q is a finite state set, Σ is the alphabet, $\delta : Q \times \Sigma \to Q$ is the state transition function, and $\tilde{\pi} : Q \times \Sigma \to [0,1]$ specifies the conditional symbol-generation probabilities. δ and $\tilde{\pi}$ are recursively extended to arbitrary $y = \sigma x \in \Sigma^\star$ as $\delta(q, \sigma x) = \delta(\delta(q,\sigma), x)$ and $\tilde{\pi}(q, \sigma x) = \tilde{\pi}(q,\sigma)\tilde{\pi}(\delta(q,\sigma), x)$. $q_0 \in Q$ is the initial state. If the next symbol is specified, our resultant state is fixed; similar to probabilistic deterministic automata [27]. However, unlike the latter, we lack final states. Additionally, we assume our graphs to be strongly connected.

Definition 2.5 has no notion of a final state, and later we will remove initial state dependence using ergodicity. First, we formalize how the notion of a PFSA arises from a QSP.

Lemma 2.6 (from QSP to a PFSA). *If the probabilistic Nerode relation has a finite index, then there exists an initial-marked PFSA generator.*

Proof. Every QSP represented as a probability space $(\Sigma^\omega, \mathbb{B}, \mu)$ induces a probabilistic automaton $(Q, \Sigma, \delta, \tilde{\pi}, q_0)$, where Q is the set of equivalence classes of the probabilistic Nerode relation (definition 2.4), Σ is the alphabet, and

$$\delta([x], \sigma) = [x\sigma] \tag{2.3}$$

and

$$\tilde{\pi}([x], \sigma) = \frac{\Pr(x'\sigma)}{\Pr(x')} \quad \text{for any choice of } x' \in [x]. \tag{2.4}$$

q_0 is identified with $[\lambda]$, and finite index of $\sim_N$ implies $|Q| < \infty$. ■

The above construction yields a *minimal realization* unique up to state renaming.

Corollary 2.7 (to lemma 2.6: null-word probability). *For the PFSA* $(Q, \Sigma, \delta, \tilde{\pi})$ *induced from a QSP* $\mathcal{H}$:

$$\forall q \in Q, \quad \tilde{\pi}(q, \lambda) = 1. \tag{2.5}$$

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

Proof. For $q \in Q$, let $x \in \Sigma^\star$ such that $[x] = q$. From equation (2.4), we have

$$\begin{aligned}\tilde{\pi}(q,\lambda) &= \frac{\Pr(x'\lambda)}{\Pr(x')} \quad \text{for } x' \in [x] \\ &= \frac{\Pr(x')}{\Pr(x')} = 1. \end{aligned} \tag{2.6}$$

■

Next, we define canonical representations to remove initial-state dependence. We use $\tilde{\Pi}$ to denote the matrix representation of $\tilde{\pi}$, i.e. $\tilde{\Pi}_{ij} = \tilde{\pi}(q_i, \sigma_j)$, $q_i \in Q, \sigma_j \in \Sigma$. We need the notion of transformation matrices Γ_σ.

Definition 2.8 (transformation matrices). For an initial-marked PFSA $G = (Q, \Sigma, \delta, \tilde{\pi}, q_0)$, the symbol-specific transformation matrices $\Gamma_\sigma \in \{0,1\}^{|Q|\times|Q|}$ are

$$\Gamma_\sigma|_{ij} = \begin{cases} \tilde{\pi}(q_i,\sigma) & \text{if } \delta(q_i,\sigma) = q_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

States in the canonical representation (denoted as $\wp_x$) are identified with probability distributions over states of the initial-marked PFSA. Here, x denotes the string in $\Sigma^\star$ realizing this distribution, beginning from the stationary distribution on the states of the initial-marked representation. $\wp_x$ is an equivalence class, and hence x is not unique.

Definition 2.9 (canonical representations). An initial-marked PFSA $G = (Q, \Sigma, \delta, \tilde{\pi}, q_0)$ uniquely induces a canonical representation $(Q^{\mathrm{C}}, \Sigma, \delta^{\mathrm{C}}, \tilde{\pi}^{\mathrm{C}})$, with Q^{C} being the set of probability distributions over Q, $\delta^{\mathrm{C}} : Q^{\mathrm{C}} \times \Sigma \to Q^{\mathrm{C}}$, and $\tilde{\pi}^{\mathrm{C}} : Q^{\mathrm{C}} \times \Sigma \to [0,1]$, as follows.

— Construct the stationary distribution on Q using the transition probabilities of the Markov chain induced by G, and include this as the first element $\wp_\lambda$ of Q^{C}. The transition matrix for this induced chain is the row-stochastic matrix $M \in [0,1]^{|Q|\times|Q|}$, with $M_{ij} = \sum_{\sigma:\delta(q_i,\sigma)=q_j} \tilde{\pi}(q_i,\sigma)$.
— Define δ^{C} and $\tilde{\pi}^{\mathrm{C}}$ recursively:

$$\delta^{\mathrm{C}}(\wp_x,\sigma) = \frac{1}{\|\wp_x\Gamma_\sigma\|_1}\wp_x\Gamma_\sigma \triangleq \wp_{x\sigma} \tag{2.8}$$

and

$$\tilde{\pi}^{\mathrm{C}}(\wp_x,\sigma) = \wp_x\tilde{\Pi}. \tag{2.9}$$

For a QSP $\mathcal{H}$, the canonical representation is denoted as $\mathcal{C}_\mathcal{H}$.

Ergodicity of QSPs, which makes $\wp_\lambda$ independent of the initial state in the initial-marked PFSA, implies that the canonical representation is initial state independent (figure 4), and subsumes the initial-marked representation in the following sense: let $\mathcal{E} = \{e^i \in [0\ 1]^{|Q|}, i = 1, \ldots, |Q|\}$ denote the set of distributions satisfying

$$e^i|_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

Then we note the following.

— If we execute the canonical construction with an initial distribution from $\mathcal{E}$, then we get the initial-marked PFSA (with the initial marking missing).
— If during the construction we encounter $\wp_x \in \mathcal{E}$ for some x, then we stay within the graph of the initial-marked PFSA for all right extensions of x. This thus eliminates the need of knowing the initial state explicitly (figure 5), provided we find string x, which takes us within or close to $\mathcal{E}$.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

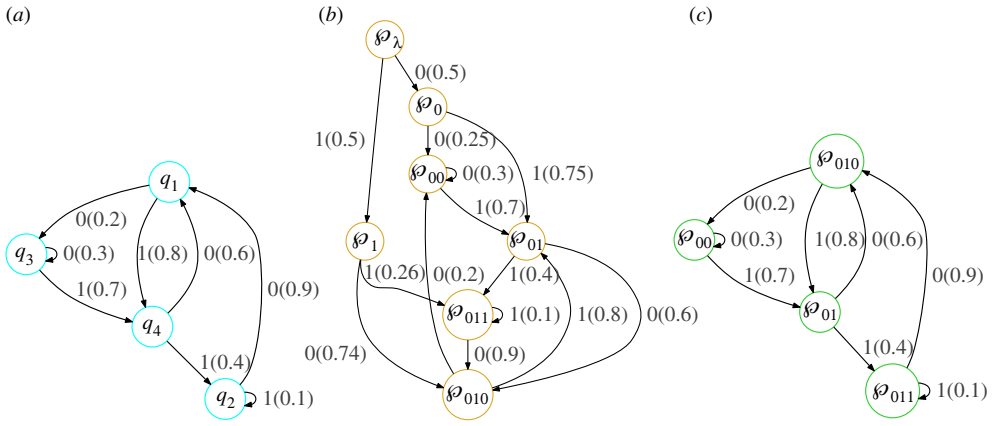


Figure 4. Illustration of canonical representations. (*a*) An initial-marked PFSA, and (*b*) The canonical representation obtained starting with the stationary distribution $\wp_\lambda$ on the states q_1, q_2, q_3 and q_4. We obtain a finite graph here, which is not guaranteed in general. (c) The strongly connected component, which is isomorphic to the original PFSA. States of the machine in (*c*) are distributions over the states of the machine in (*a*). (Online version in colour.)

Consequently, we denote the initial-marked PFSA induced by a QSP $\mathcal{H}$, with the initial marking removed, as $\mathcal{P}_\mathcal{H}$, and refer to it simply as a 'PFSA' (dropping the qualifier 'initial-marked'). States in $\mathcal{P}_\mathcal{H}$ are representable as states in $\mathcal{C}_\mathcal{H}$ as elements of $\mathcal{E}$. Next, we establish a key result: we always encounter a state arbitrarily close to some element in $\mathcal{E}$ in the canonical construction starting from the stationary distribution $\wp_\lambda$.

Theorem 2.10 (ϵ-synchronization of probabilistic automata). *For any QSP $\mathcal{H}$ over Σ, the PFSA $\mathcal{P}_\mathcal{H}$ satisfies*

$$\forall \epsilon' > 0, \quad \exists x \in \Sigma^\star, \quad \exists \vartheta \in \mathcal{E}, \quad \|\wp_x - \vartheta\| \leqq \epsilon', \tag{2.11}$$

where the norm used is unimportant.

Proof. See appendix A. ■

Theorem 2.10 induces the notion of ϵ-synchronizing strings, and guarantees their existence for arbitrary PFSA.

Definition 2.11 (ϵ-synchronizing strings). An ϵ-synchronizing string $x \in \Sigma^\star$ for a PFSA is one that satisfies

$$\exists \vartheta \in \mathcal{E}, \quad \|\wp_x - \vartheta\| \leqq \epsilon. \tag{2.12}$$

The norm used is unimportant.

Theorem 2.10 does not yield an algorithm for computing synchronizing strings (theorem 2.18). It simply shows that one always exists. As a corollary, we estimate an asymptotic upper bound on the effort required to find it.

Corollary 2.12 (to theorem 2.10). *At most $O(1/\epsilon)$ strings need to be analysed to find an ϵ-synchronizing string.*

Proof. See appendix A. ■

Next, we describe the basic principle of our inference algorithm. PFSA states are not observable; we observe symbols generated from hidden states. This leads us to the notion of *symbolic derivatives*, which are computable from observations.

We denote the set of probability distributions over a set of cardinality k as $\mathcal{D}(k)$. First, we specify a count function.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

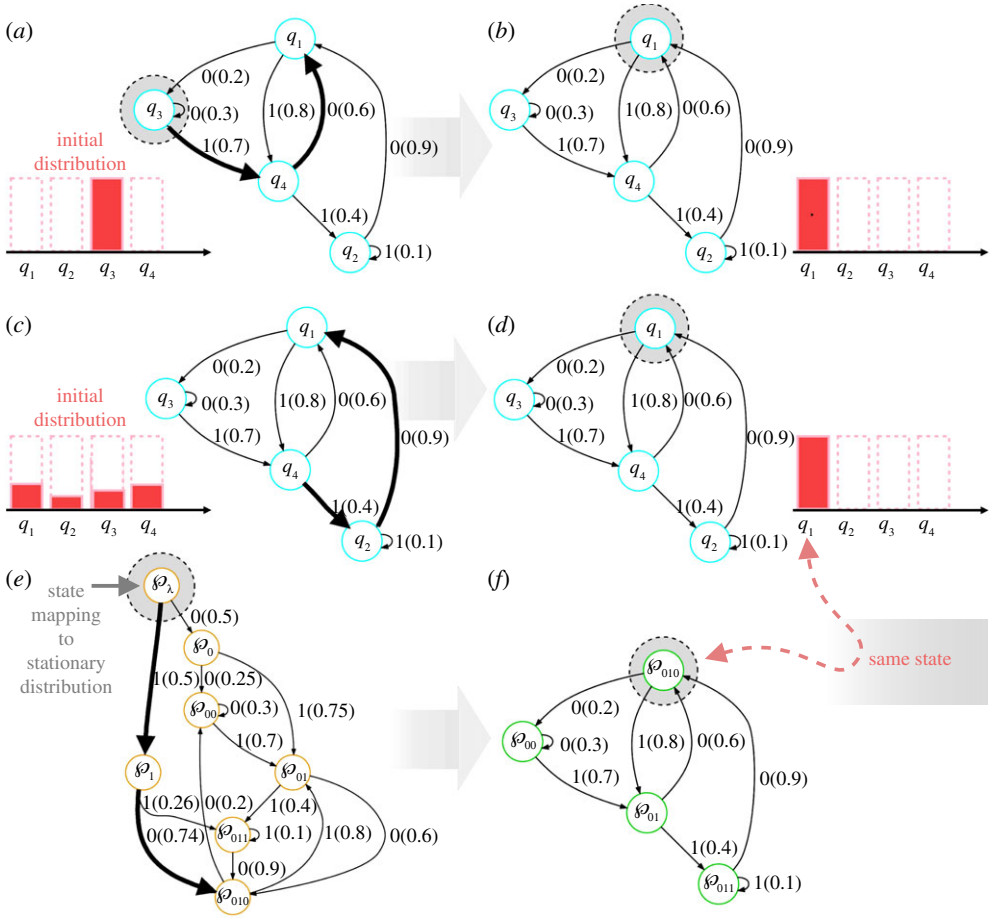


Figure 5. Why initial states are unnecessary. A sequence 10 is executed in the initial-marked PFSA in (*a*) from q_3 (transitions shown in bold), resulting in the new state q_1 (*b*). For this model, the same final state would be obtained by executing 10 from any other initial state, and also from any initial distribution over the states (as shown in (*c*) and (*d*)). This is because 10 is a synchronizing string for this model. Thus, if the initial state is unknown, and we assume that the initial distribution is stationary, then we actually start from the state $\wp_\lambda$ in the canonical representation (*e*), and we would end up with a distribution that has support on a unique state in the initial-marked machine (see (*f*), where the resulting state in the pruned representation corresponds to q_1 in the initial-marked machine). Knowing a synchronizing string (of sufficient length) therefore makes the initial condition irrelevant. This machine is perfectly synchronizable, which is not always true, e.g. model M2 in figure 3. However, approximate synchronization is always achievable (theorem 2.10), which therefore eliminates the need of knowing the initial state explicitly, but requires us to compute such an ϵ-synchronizing sequence. (Online version in colour.)

Definition 2.13 (symbolic count function). For a string s over Σ, the count function $\#^s : \Sigma^\star \to \mathbb{N} \cup \{0\}$ counts the number of times a particular substring occurs in s. The count is overlapping, i.e. in a string $s = 0001$, we count the number of occurrences of 00s as $\underline{00}01$ and $0\underline{00}1$, implying $\#^s 00 = 2$.

Definition 2.14 (symbolic derivative). For a string s generated by a QSP over Σ, the symbolic derivative is a function $\phi^s : \Sigma^\star \to \mathcal{D}(|\Sigma| - 1)$ as

$$\phi^s(x)|_i = \frac{\#^s x\sigma_i}{\sum_{\sigma_i \in \Sigma} \#^s x\sigma_i}. \tag{2.13}$$

Thus, $\forall x \in \Sigma^\star, \phi^s(x)$ is a probability distribution over Σ. $\phi^s(x)$ is referred to as the symbolic derivative at x.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

For $q_i \in Q$, $\tilde{\pi}$ induces a distribution over Σ as $[\tilde{\pi}(q_i, \sigma_1), \ldots, \tilde{\pi}(q_i, \sigma_{|\Sigma|})]$. We denote this as $\tilde{\pi}(q_i, \cdot)$. We show that the symbolic derivative at x can be used to estimate this distribution for $q_i = [x]$, provided x is ϵ-synchronizing.

Theorem 2.15 (ϵ-convergence). *If $x \in \Sigma^\star$ is ϵ-synchronizing then*

$$\forall \epsilon > 0, \quad \lim_{|s| \to \infty} \|\phi^s(x) - \tilde{\pi}([x], \cdot)\|_\infty \leqq_{\text{a.s.}} \epsilon. \tag{2.14}$$

Proof. The proof follows from the Glivenko–Cantelli theorem [31] on uniform convergence of empirical distributions. See appendix A for details. ■

Next, we describe identification of ϵ-synchronizing strings given a sufficiently long observed string s. Theorem 2.10 guarantees existence, and corollary 2.12 establishes that $O(1/\epsilon)$ substrings need to be analysed until we encounter an ϵ-synchronizing string. These do not provide an executable algorithm, which arises from an inspection of the geometric structure of the set of probability vectors over Σ, obtained by constructing $\phi^s(x)$ for different choices of the candidate string x.

Definition 2.16 (derivative heap). Given a string s generated by a QSP, a derivative heap $\mathcal{D}^s : 2^{\Sigma^\star} \to \mathcal{D}(|\Sigma| - 1)$ is the set of probability distributions over Σ calculated for a given subset of strings $L \subset \Sigma^\star$ as follows:

$$\mathcal{D}^s(L) = \{\phi^s(x) : x \in L \subset \Sigma^\star\}. \tag{2.15}$$

Lemma 2.17 (limiting geometry). *Let $\mathcal{D}_\infty = \lim_{|s| \to \infty} \lim_{L \to \Sigma^\star} \mathcal{D}^s(L)$, and $\mathcal{U}_\infty$ be the convex hull of $\mathcal{D}_\infty$. If u is a vertex of $\mathcal{U}_\infty$, then*

$$\exists q \in Q \text{ such that } u = \tilde{\pi}(q, \cdot). \tag{2.16}$$

Proof. Recalling theorem 2.15, the result follows from noting that any element of $\mathcal{D}_\infty$ is a convex combination of elements from the set $\{\tilde{\pi}(q_1, \cdot), \ldots, \tilde{\pi}(q_{|Q|}, \cdot)\}$. ■

Lemma 2.17 does not claim that the number of vertices of the convex hull of $\mathbb{D}_\infty$ equals the number of states, but that every vertex corresponds to a state. We cannot generate $\mathcal{D}_\infty$ in practice since we have a finite observed string s, and we can only calculate $\phi^s(x)$ for a finite number of x. Instead, we show that choosing a string corresponding to the vertex of the convex hull of the heap, constructed by considering $O(1/\epsilon)$ strings, gives us an ϵ-synchronizing string with high probability.

Theorem 2.18 (derivative heap approximation). *For s generated by a QSP, let $\mathcal{D}^s(L)$ be computed with $L = \Sigma^{O(\log(1/\epsilon))}$. If for some $x_0 \in \Sigma^{O(\log(1/\epsilon))}$, $\phi^s(x_0)$ is a vertex of the convex hull of $\mathcal{D}^s(L)$, then*

$$\text{Prob}(x_0 \text{ is not } \epsilon\text{-synchronizing}) \leqq e^{-|s|\epsilon O(1)}. \tag{2.17}$$

Proof. The result follows from Sanov's theorem [32] for convex set of probability distributions. See appendix A for details. ■

Figure 6 illustrates PFSAs, derivative heaps, and ϵ-synchronizing strings. Next, we present our inference algorithm.

3. Algorithm GenESeSS

(a) Implementation steps

We call our algorithm 'Generator Extraction Using Self-similar Semantics', or GenESeSS which for an observed sequence s, consists of three steps.

— *Identification of ϵ-synchronizing string x_0.* Construct a derivative heap $\mathcal{D}^s(L)$ using the observed trace s (definition 2.16), and set L consisting of all strings up to a sufficiently large, but finite, depth. We suggest as initial choice of L as $\log_{|\Sigma|} 1/\epsilon$. If L is sufficiently

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

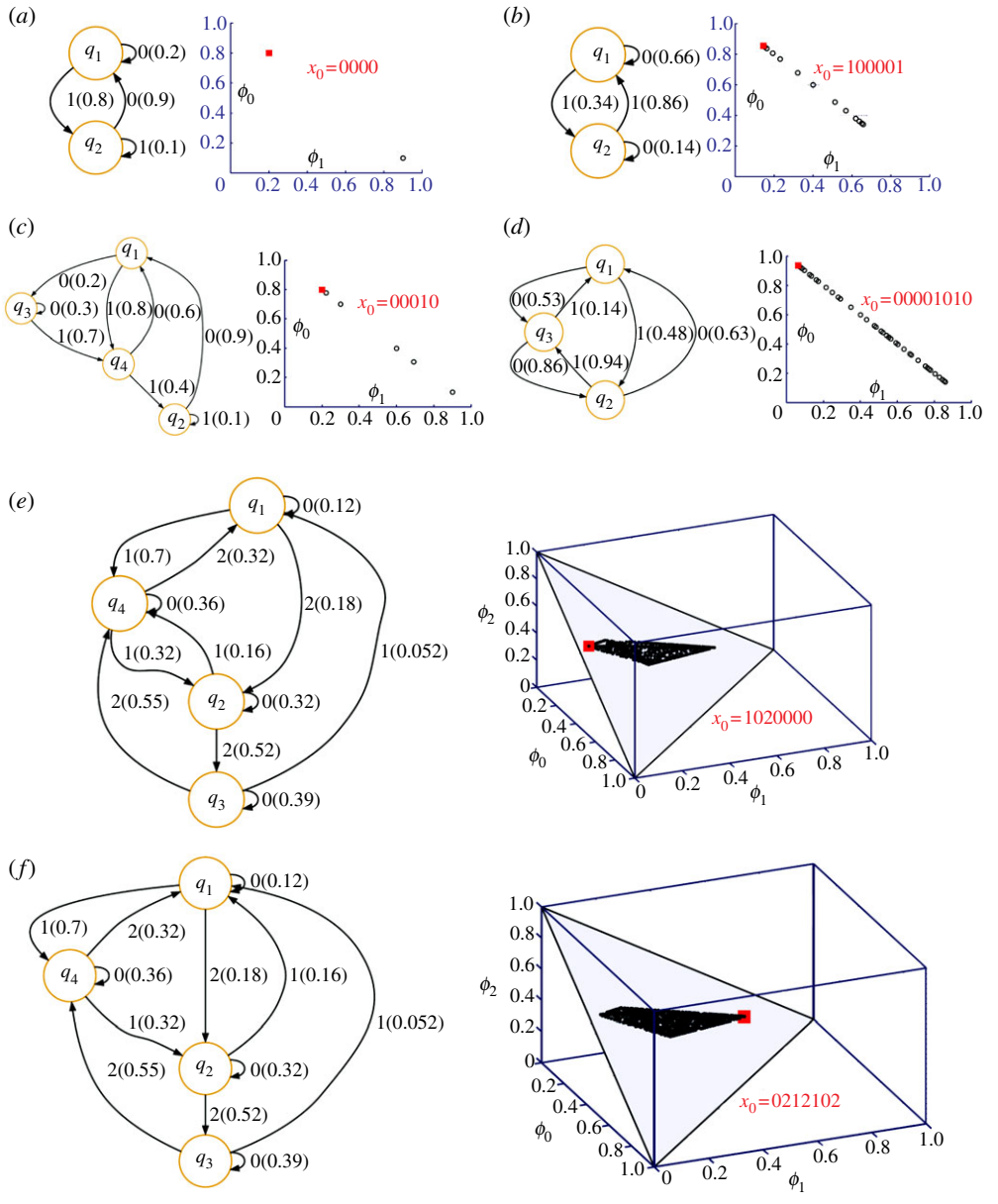


Figure 6. Illustration of derivative heaps. (*a*, *c*) Synchronizable PFSAs, and hence the derivative heaps have few distinct images (two for (*a*), and about five for (*c*)), while the PFSAs shown in (*b*) and (*d*) are non-synchronizable, and their heaps consist of a spread of points. Note since the alphabet size is 2 in these cases, the heaps are part of a line segment. (*e*,*f*) PFSAs and derivative heaps for 3-letter alphabets. The derivative heap is a polytope as expected. Red square marks the derivative at the chosen ϵ-synchronizing string x_0 in all cases. ϕ_i is the *i*th coordinate of the symbolic derivative. (Online version in colour.)

large, then the inferred model structure will not change for larger values. We then identify a vertex of the convex hull for $\mathcal{D}_\infty$, via any standard algorithm for computing the hull [33]. Choose x_0 as the string mapping to this vertex.

— *Identification of the structure of $\mathcal{P}_\mathcal{H}$, i.e. transition function δ.* We generate δ as follows. For each state q, we associate a string identifier $x_q^{\mathrm{id}} \in x_0\Sigma^\star$, and a probability distribution h_q

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

on Σ (which is an approximation of the $\tilde{\Pi}$-row corresponding to state q). We extend the structure recursively:

(i) Initialize the set Q as $Q = \{q_0\}$, and set $x^{\mathrm{id}}_{q_0} = x_0$, $h_q = \phi^s(x_0)$.
(ii) For each state $q \in Q$, compute for each symbol $\sigma \in \Sigma$, find symbolic derivative $\phi^s(x^{\mathrm{id}}_q \sigma)$. If $\|\phi^s(x^{\mathrm{id}}_q \sigma) - h_{q'\sigma}\|_\infty \leqq \epsilon$ for some $q' \in Q$, then define $\delta(q, \sigma) = q'$. If, on the other hand, no such q' can be found in Q, then add a new state q' to Q, and define $x^{\mathrm{id}}_{q'} = x^{\mathrm{id}}_q \sigma$, $h_{q'} = \phi^s(x^{\mathrm{id}}_q \sigma)$.

The process terminates when every $q \in Q$ has a target state, for each $\sigma \in \Sigma$. Then, if necessary, we ensure strong connectivity using [34].

— *Identification of arc probabilities*, i.e. *function* $\tilde{\pi}$*:*

(i) Choose an arbitrary initial state $q \in Q$.
(ii) Run sequence s through the identified graph, as directed by δ, i.e. if current state is q, and the next symbol read from s is σ, then move to $\delta(q, \sigma)$. Count arc traversals, i.e. generate numbers N^i_j where $q_i \xrightarrow[N^i_j]{\sigma_j} q_k$.
(iii) Generate $\tilde{\Pi}$ by row normalization, i.e. $\tilde{\Pi}_{ij} = N^i_j / (\sum_j N^i_j)$.

Gavaldà *et al*. [27] and Castro & Gavaldà [28] use similar recursive structure extension. However, with no notion of ϵ-synchronization, they are restricted to inferring only synchronizable or short-memory models, or large approximations for long-memory ones (figure 7).

(b) Complexity analysis and probably approximately correct learnability

While h_q (in step 2) approximates $\tilde{\Pi}$ rows, we find the arc probabilities via normalization of traversal count. h_q only uses sequences in $x_0 \Sigma^\star$, while traversal counting uses the entire sequence s, and is more accurate.

GenESeSS has no upper bound on the number of states; which is a function of the complexity of the process itself.

Theorem 3.1 (time complexity). *Asymptotic time complexity of* GenESeSS *is*

$$\mathcal{T} = O\left(\left(\frac{1}{\epsilon} + |Q|\right) \times |s| \times |\Sigma|\right). \tag{3.1}$$

Proof. See appendix A for details. ■

Theorem 3.1 shows that GenESeSS is polynomial in $O(1/\epsilon)$, size of input s, model size $|Q|$ and alphabet size $|\Sigma|$. In practice, $|Q| \ll 1/\epsilon$, implying that

$$\mathcal{T} = O\left(\frac{|s||\Sigma|}{\epsilon}\right). \tag{3.2}$$

An identification method is said to identify a target language $L_\star$ in the PAC sense [29,35,36] if it always halts and outputs L such that

$$\exists \epsilon,\ \delta > 0,\ P(d(L_\star, L) \leqq \epsilon) \geqq 1 - \delta, \tag{3.3}$$

where $d(\cdot, \cdot)$ is a metric on the space of target languages. A class of languages is efficiently PAC-learnable if there exists an algorithm that PAC-identifies every language in the class, and runs in time polynomial in $1/\epsilon$, $1/\delta$, the length of sample input, and inferred model size. We prove PAC-learnability of QSPs by first establishing a metric on the space of probabilistic automata over Σ.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

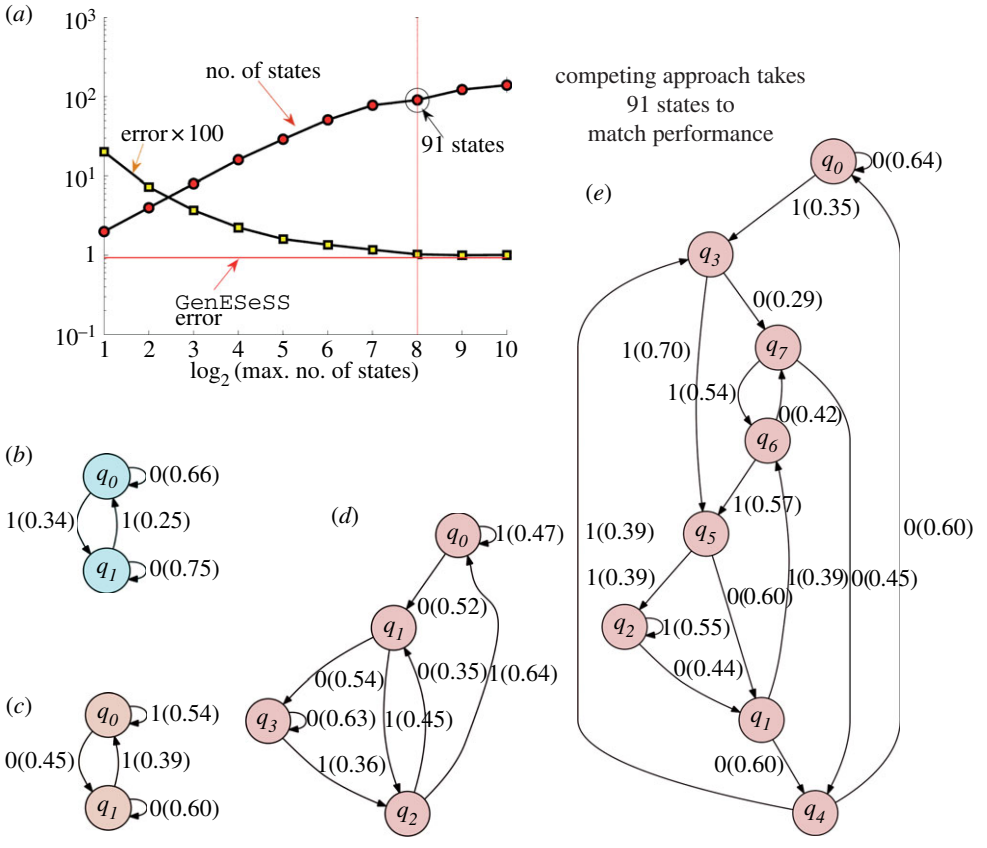


Figure 7. Application of the competing algorithm of [27,28] on simulated data from model M2 (figure 3) with $\epsilon = 0.05$. (*a*) The modelling error in the metric defined in lemma 3.2 with increasing model size, requiring 91 states to match GenESeSS performance (which finds the correct 2 state model M2 (*b*)). Gavaldà *et al.* requires a maximum model size, and (*c*)–(*e*) show models obtained with 2, 4 and 8 states. All models found are synchronizable, and miss the key point found by GenESeSS that M2 is non-synchronizable with LRDs. Algorithm CSSR [12] also finds only similar synchronizable models. (Online version in colour.)

(c) Probably approximately correct identifiability of quantized stochastic processes

Lemma 3.2 (metric for probabilistic automata). *For two strongly connected PFSAs G_1 and G_2, denote the symbolic derivative at $x \in \Sigma^\star$ as $\phi^s_{G_1}(x)$ and $\phi^s_{G_2}(x)$, respectively. Then,*

$$\Theta(G_1, G_2) = \sup_{x \in \Sigma^\star} \left\{ J(x) = \lim_{|s_1| \to \infty} \lim_{|s_2| \to \infty} \|\phi^{s_1}_{G_1}(x) - \phi^{s_2}_{G_2}(x)\|_\infty \right\}$$

defines a metric on the space of probabilistic automata on Σ.

Proof. Non-negativity and symmetry follow immediately. Triangular inequality follows from noting that $\|\phi^{s_1}_{G_1}(x) - \phi^{s_2}_{G_2}(x)\|_\infty$ is upper bounded by 1, and therefore for any chosen order of the strings in $\Sigma^\star$, we have two ℓ_∞ sequences, which would satisfy the triangular inequality under the sup norm. The metric is well defined since for any sufficiently long s_1 and s_2, the symbolic derivatives at arbitrary x are uniformly convergent to some linear combination of the rows of the corresponding $\tilde{\Pi}$ matrices. ■

Theorem 3.3 (PAC-learnability). *QSPs for which the probabilistic Nerode equivalence has a finite index are PAC-learnable using PFSAs, i.e. for $\epsilon, \eta > 0$, and for every sufficiently long sequence s generated*

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

by QSP $\mathcal{H}$, we can compute $\mathcal{P}'_{\mathcal{H}}$ as an estimate for $\mathcal{P}_{\mathcal{H}}$ such that

$$\text{Prob}(\Theta(\mathcal{P}_{\mathcal{H}}, \mathcal{P}'_{\mathcal{H}}) \leqq \epsilon) \geqq 1 - \eta. \tag{3.4}$$

The algorithm runs in time polynomial in $1/\epsilon, 1/\eta$, input length $|s|$ and model size.

Proof. GenESeSS construction implies that, once the initial ϵ-synchronizing string x_0 is identified, there is no scope of the model error to be more than ϵ. Hence

$$\begin{aligned}\text{Prob}(\Theta(\mathcal{P}_{\mathcal{H}}, \mathcal{P}'_{\mathcal{H}}) \leqq \epsilon) = 1 - \text{Prob}(\|\phi^s(x_0) - \wp_{x_0}\tilde{\Pi}\|_\infty > \epsilon)\\ \Rightarrow \text{Prob}(\Theta(\mathcal{P}_{\mathcal{H}}, \mathcal{P}'_{\mathcal{H}}) \leqq \epsilon) \geqq 1 - \mathrm{e}^{-|s|\epsilon O(1)} \quad \text{(using equation (A 12)).}\end{aligned}$$

Thus, for any $\eta > 0$, if we have $|s| = O(1/\epsilon \log 1/\eta)$, then the required condition of equation (3.4) is met. The polynomial runtimes are established in theorem 3.1. ■

(i) Remark on Kearns' hardness result

We are immune to Kearns' hardness result [37], since $\epsilon > 0$ enforces state distinguishability [16].

4. Discussion, validation and application

(a) Application of GenESeSS to simulated data

For the scenario in figure 3, GenESeSS is applied to 10^3 symbols generated from the model M2 (denoting σ_0 as 0, and σ_1 as 1) with parameters $\gamma = 0.75$ and $\gamma' = 0.66$. We generate a derivative heap with all strings of length up to 4, and find 000 mapping to an estimated vertex of the convex hull of the heap, thus qualifying as our ϵ-synchronizing string x_0. Figure 8*a*,*b* illustrate the choice of a 'wrong' x_0; our theoretical development guarantees that if x_0 is close to a vertex of the convex hull, then the derivatives $\phi^s(x_0x)$ will be close to some row of $\tilde{\Pi}$, and thus form clusters around those values. If the chosen x_0 is far from all vertices, this is not true; thus, in figure 8*a*, the incorrect choice of string 11 leads to a spread of symbolic derivatives on the line $x + y = 1$. Figure 8*b* illustrates clustering of the values around the two hidden states with $x_0 = 000$. A few symbolic derivatives are shown in figure 8*c*, with highlighted rows corresponding to strings with x_0 as prefix; which are approximate repetitions of two distinct distributions. Switching between these repeating distributions on right concatenation of symbols induces the structure shown in figure 8*d* for $\epsilon = 0.05$ (see §3*a*). The inferred model is already strongly connected, and we run the input data through it (step 3 of GenESeSS) to estimate the $\tilde{\Pi}$ matrix (see the inset of figure 8*d*). The model is recovered with correct structure, and with deviation in $\tilde{\Pi}$ entries smaller than 0.01.

This example infers a non-synchronizable machine (recall that M2 exhibits LRDs), which often proves to be too difficult for reported unsupervised algorithms (see comparison with [27] in figure 7, which can match GenESeSS performance only at the expense of a large number of states, and yet still fails to infer true LRD models). Hidden Markov model learning with latent transitions can possibly be applied; however, such algorithms do not come with any obvious guarantees on performance.

(b) Application of GenESeSS to model and predict experimental data

We apply GenESeSS to two experimental datasets: photometric intensity (brightness) of a variable star for a succession of 600 days [38], and twice-daily count of bow-fly population in a jar [39]. While the photometric data are quasi-periodic, the population time series has non-trivial stochastic structure. Quantization schemes, with four symbols, are shown in table 1.

The mean magnitude of successive change represented by each symbol is also shown, which is computed as the average of the continuous values that get mapped to each symbol, e.g. for the photometric data, symbol 0 represents a change of brightness in the range $[-4, -2]$, and an

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

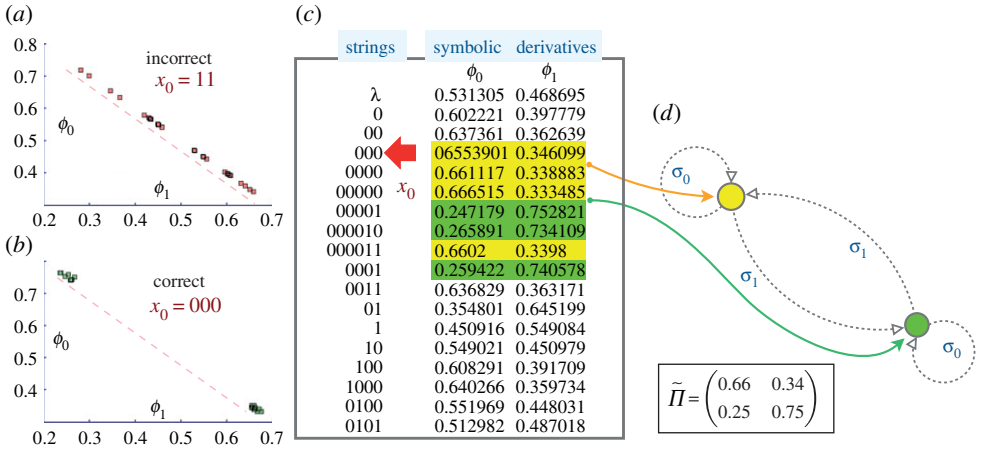


Figure 8. Application of GenESeSS to simulated data for highway traffic scenario (model M2, see figure 3). (*a*,*b*) The correct and wrong choices of ϵ-synchronizing x_0. The derivative heap for right extensions of correctly chosen x_0 has few distinct images (*b*). (*c*) Some symbolic derivatives, and the chosen x_0 is marked with a red arrow. Right extensions of $x_0 = 000$ fall into two classes (shown in yellow and green), and switching between them induces the structure in (*d*). Finally, $\tilde{\Pi}$ (inset of (*d*)) is estimated by step 3 of GenESeSS. (Online version in colour.)

Table 1. Quantization scheme and inferred model parameters.

system	symbol	range of change between observations	mean change represented	inferred PFSA	comparative ARMAX model order
bow-fly population in a jar	0	[−1500, −200]	−540.68	no. of states: 9	(15,5)
	1	[−200, 0]	−87.66		
	2	[0, 200]	70.4		
	3	[200, 1500]	507.12		
photometry from variable star	0	[−4, −2]	−2.67	no. of states: 8	(3,1)
	1	[−2, 0]	−0.44		
	2	[0, 2]	1.35		
	3	[2, 4]	3.12		

average change of −2.67. Inferred PFSAs generate symbolic sequences, which we map back to the continuous domain using these average magnitudes, e.g. a generated 0 is interpreted as a change of −2.67 for photometric predictions.

The inferred PFSAs are shown in figures 9*b* and 10*b*. The causal structure is simple for the photometric data, and significantly more complex for the population dynamics. For the photometric dataset, the states with only outgoing symbols 2,3 (indicating a brightness increase next day), the states with outgoing symbols 1,2 (indicating the possibility of either an increase or a decrease), and the states with outgoing symbols 2,3 (indicating a decrease in intensity next day) are arranged in a simple succession (figure 9*b*). For the population dynamics, the group of states which indicate a predicted population increase in the next 12 h, the ones which indicate a decrease, and the ones which indicate either possibility have significantly more complex interconnections (figure 10*c*). In both cases, we learn ARMAX [7] models for comparison. ARMAX models include auto-regressive and moving average components, assume additive Gaussian noise, and linear dependence on history. ARMAX model order (p, q) is chosen to be the smallest value-pair providing an acceptable fit to the data (table 1), such that increasing the model order does not significantly improve prediction accuracy. We learn the respective

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

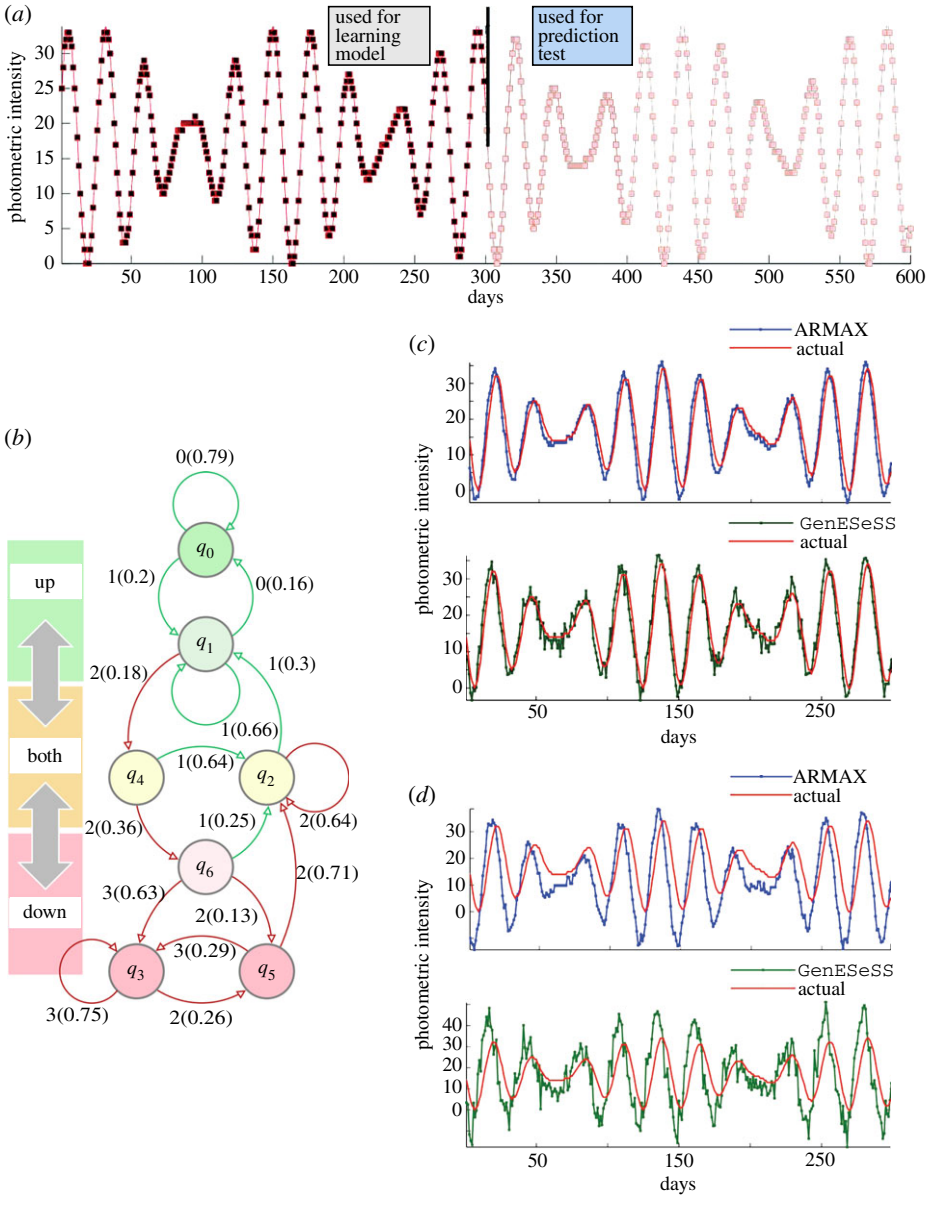


Figure 9. Modelling photometric intensity of a variable star (source: Newton [38]). (*a*) Time-series data, (*b*) inferred PFSA. (*c*,*d*) Predicted series for GenESeSS and ARMAX model. (Online version in colour.)

models with a portion of the time series (figures 9*a* and 10*a*), and compare the approaches with different prediction horizons ('prediction horizon = 1 day' means 'predict values 1 day ahead') against the remaining part of the time series. ARMAX does well in the first case, even for longer prediction horizons (see figure 9*c*,*d*), but is unacceptably poor with the more involved population dynamics (figure 10*d*,*e*). Additionally, the ARMAX models do not offer clear insight into the causal structure of the processes as discussed above. Figure 11*a*,*c* plots the mean absolute prediction error percentage as a function of increasing prediction horizon for both approaches. For the photometric data, the error grows slowly, and is more or less comparable for ARMAX and GenESeSS. For the population data, GenESeSS does significantly better. Since we generate continuous prediction series from symbolic sequences using mean magnitudes (as described above; see also table 1, column 4), GenESeSS is expected to better predict the sign of the gradient

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

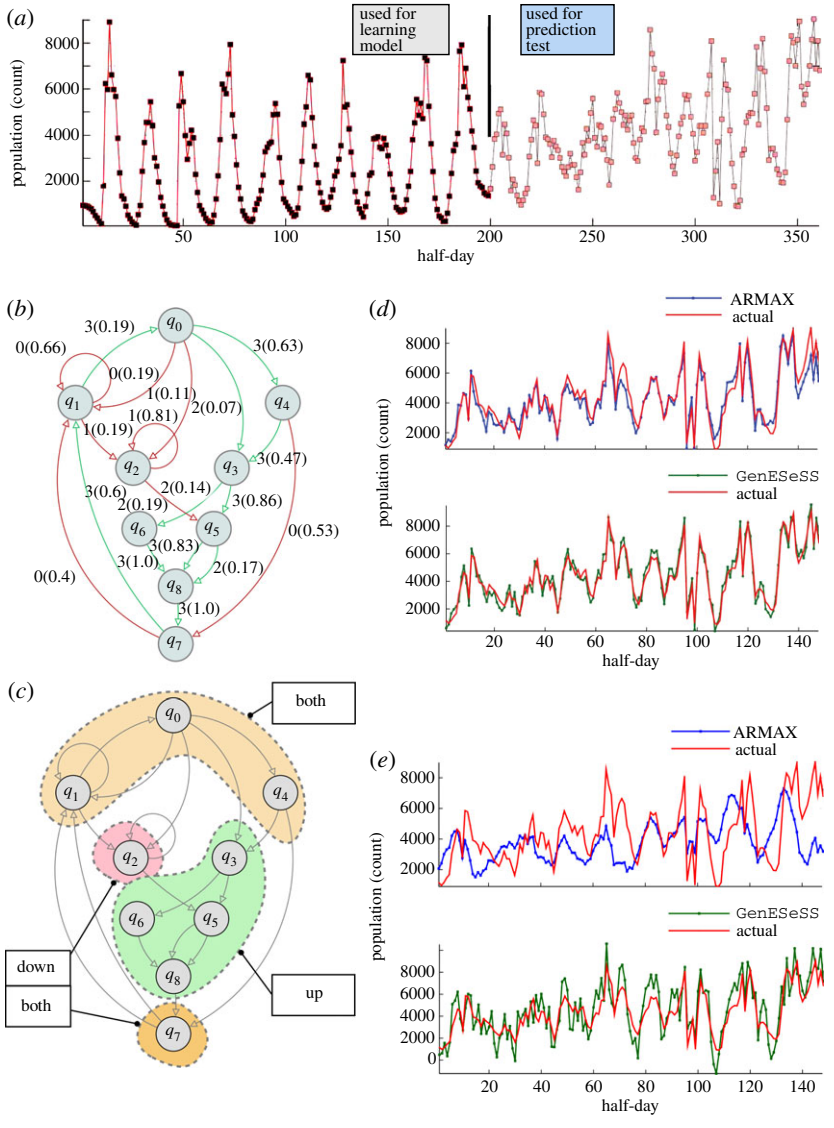


Figure 10. Modelling bow-fly population in a jar (source: Nicholson [39]). (*a*) Time-series data, (*b*) inferred PFSA. (*c*) The interconnection between groups of states that indicate population increase (only symbols 2 and 3 defined), decrease (only symbols 0 and 1 defined), or both in the next observed measurement. (*d*,*e*) Predicted series for GenESeSS and ARMAX model. (Online version in colour.)

of the predicted data rather than the exact magnitudes. We plot the percentage of wrongly predicted trends (i.e. percentage of sign mismatches) in figure 11*b*,*d* for the two cases. We note that ARMAX is slightly better for the photometric data, while for the population data, GenESeSS is significantly superior (75% incorrect trends at a horizon of 8 days for ARMAX versus about 25% for GenESeSS). Thus, while for quasi-periodic data GenESeSS performance may be comparable to existing techniques, it is significantly superior in modelling and predicting complex stochastic dynamics.

5. Summary and conclusion

We establish the notion of causal generators for stationary ergodic QSPs on a rigorous measure-theoretic foundation, and propose a new inference algorithm GenESeSS for

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

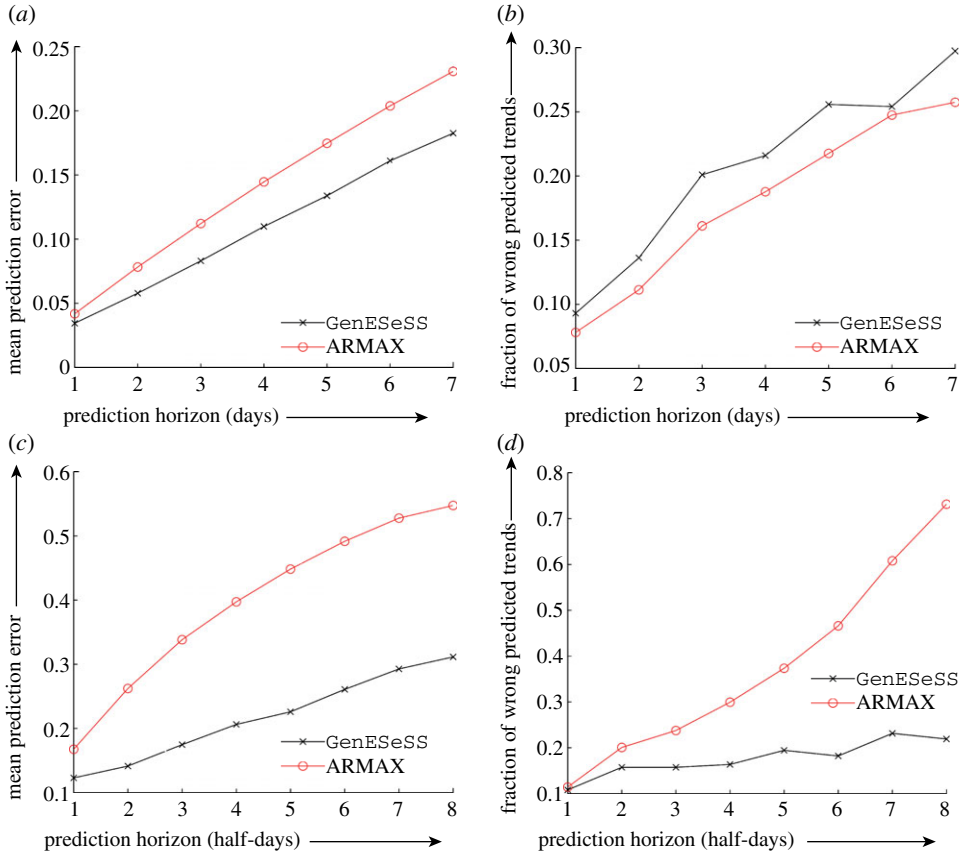


Figure 11. Comparison of GenESeSS against ARMAX model (*a*,*b*) photometric and (*c*,*d*) population time series. (*a*,*c*) Variation of the mean absolute prediction error as a function of the prediction horizon. (*b*,*d*) Variation of the percentage of wrongly predicted trends, i.e. the number of times the sign of the predicted gradient failed to match up with the ground truth. (Online version in colour.)

identification of probabilistic automata models from sufficiently long traces. We show that our approach can learn processes with LRDs, which yield non-synchronizable automata. Additionally, we establish that GenESeSS is computationally efficient in the PAC sense. The results are validated on synthetic and real datasets. Future research will investigate the prediction problem in greater detail in the context of new applications.

This work was supported in part by NSF CDI grant ECCS 0941561 and DTRA grant HDTRA 1-09-1-0013. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the sponsoring organizations.

Appendix A. Detailed proofs of mathematical results

We present the proofs of theorems 2.10, 2.15, 2.18, 3.1, and for corollary 2.12 to theorem 2.10.

— (*Statement of theorem* 2.10) *For any QSP* $\mathcal{H}$ *over* Σ, *the PFSA* $\mathcal{P}_{\mathcal{H}}$ *satisfies*

$$\forall \epsilon' > 0, \quad \exists x \in \Sigma^\star, \quad \exists \vartheta \in \mathcal{E}, \quad \|\wp_x - \vartheta\| \leqq \epsilon',$$

where the norm used is unimportant.

Proof. We show that all PFSA are at least approximately synchronizable [40,41], which is not true for deterministic automata. If the graph of $\mathcal{P}_{\mathcal{H}}$ (i.e. the deterministic automaton obtained by

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

removing the arc probabilities) is synchronizable, then equation (5) trivially holds true for $\epsilon' = 0$ for any synchronizing string x. Thus, we assume the graph of $\mathcal{P}_\mathcal{H}$ to be non-synchronizable. From the definition of non-synchronizability, it follows:

$$\forall q_i, q_j \in Q \quad \text{with } q_i \neq q_j, \quad \forall x \in \Sigma^\star, \quad \delta(q_i, x) \neq \delta(q_j, x). \tag{A 1}$$

If the PFSA has a single state, then every string satisfies the condition in equation (5). Hence, we assume that the PFSA has more than one state. Now if we have

$$\forall x \in \Sigma^\star, \quad \frac{\Pr(x'x)}{\Pr(x')} = \frac{\Pr(x''x)}{\Pr(x'')} \quad \text{where } [x'] = q_i, [x''] = q_j, \tag{A 2}$$

then, by definition 2.4, we have a contradiction $q_i = q_j$. Hence, $\exists x_0$ such that

$$\frac{\Pr(x'x_0)}{\Pr(x')} \neq \frac{\Pr(x''x_0)}{\Pr(x'')} \quad \text{where } [x'] = q_i, [x''] = q_j. \tag{A 3}$$

Since

$$\sum_{x \in \Sigma^\star} \frac{\Pr(x'x)}{\Pr(x')} = 1 \quad \text{for any } x' \text{ where } [x'] = q_i, \tag{A 4}$$

we conclude without loss of generality that $\forall q_i, q_j \in Q$, with $q_i \neq q_j$:

$$\exists x^{ij} \in \Sigma^\star, \quad \frac{\Pr(x'x^{ij})}{\Pr(x')} > \frac{\Pr(x''x^{ij})}{\Pr(x'')} \quad \text{where } [x'] = q_i, [x''] = q_j.$$

It follows from induction that if we start with a distribution $\wp$ on Q such that $\wp_i = \wp_j = 0.5$, then for any $\epsilon' > 0$ we can construct a finite string x_0^{ij} such that if $\delta(q_i, x_0^{ij}) = q_r, \delta(q_j, x_0^{ij}) = q_s$, then for the new distribution $\wp'$ after execution of x_0^{ij} will satisfy $\wp'_s > 1 - \epsilon'$. Recalling that $\mathcal{P}_\mathcal{H}$ is strongly connected, for any $q_t \in Q$, there exists a string $y \in \Sigma^\star$, such that $\delta(q_s, y) = q_t$. Setting $x_\star^{i,j \to t} = x_0^{ij} y$, we can ensure that the distribution $\wp''$ obtained after execution of $x_\star^{ij}$ satisfies $\wp''_t > 1 - \epsilon'$ for any q_t of our choice. For arbitrary initial distributions $\wp^A$ on Q, we consider contributions arising from simultaneously executing $x_\star^{i,j \to t}$ from states other than just q_i and q_j. Nevertheless, it is easy to see that executing $x_\star^{i,j \to t}$ implies that in the new distribution $\wp^{A'}$, we have $\wp_t^{A'} > \wp_i^A + \wp_j^A - \epsilon'$. It immediately follows that executing of the string $x^{1,2 \to |Q|} x^{3,4 \to |Q|} \dots x^{n-1, n \to |Q|}$, where

$$n = \begin{cases} |Q| & \text{if } |Q| \text{is even,} \\ |Q| - 1 & \text{otherwise} \end{cases} \tag{A 5}$$

would result in a final distribution $\wp^{A''}$ which satisfies $\wp_{|Q|}^{A''} > 1 - \frac{1}{2} n \epsilon'$. Appropriate scaling of ϵ' then completes the proof. ■

— (*Statement of corollary 2.12 to theorem 2.10*) *At most* $O(1/\epsilon)$ *strings need to be analysed to find an* ϵ*-synchronizing string*.

Proof. Theorem 2.10 works by multiplying entries from the $\tilde{\Pi}$ matrix, which cannot be all identical (otherwise the states would collapse). Let the minimum difference between two unequal entries be η. Then, following the construction in theorem 2.10, the length ℓ of the synchronizing string, up to linear scaling, satisfies $\eta^\ell = O(\epsilon)$, implying $\ell = O(\log(1/\epsilon))$. The number of strings to be analysed therefore is at most all strings of length ℓ, which is given by

$$|\Sigma|^\ell = |\Sigma|^{O(\log(1/\epsilon))} = O(1/\epsilon). \tag{A 6}$$

This completes the proof. ■

— (*Statement of theorem 2.15 on* ϵ*-convergence*) *If* $x \in \Sigma^\star$ *is* ϵ*-synchronizing, then*

$$\forall \epsilon > 0 \quad \lim_{|s| \to \infty} \| \phi^s(x) - \tilde{\pi}([x], \cdot) \|_\infty \leqq_{\text{a.s.}} \epsilon. \tag{A 7}$$

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

Proof. The proof follows from the Glivenko–Cantelli theorem [31] on uniform convergence of empirical distributions. Since x is ϵ-synchronizing, we have

$$\forall \epsilon > 0, \quad \exists \vartheta \in \mathcal{E}, \quad \|\wp_x - \vartheta\| \leqq \epsilon. \tag{A 8}$$

Let x ϵ-synchronize to q. Thus, every time we encounter x while reading s, we are guaranteed to be distributed over Q, and at most ϵ distance from the element of $\mathcal{E}$ corresponding to q. Assuming that we encounter $n(|s|)$ occurrences of x within s, we note that $\phi^s(x)$ is an approximate empirical distribution for $\tilde{\pi}(q, \cdot)$. Denoting $F_{n(|s|)}$ as the perfect empirical distributions (i.e. ones that would be obtained for $\epsilon = 0$), we have

$$\begin{aligned}
&\lim_{|s|\to\infty} \|\phi^s(x) - \tilde{\pi}(q, \cdot)\|_\infty \\
&\quad = \lim_{|s|\to\infty} \|\phi^s(x) - F_{n(|s|)} + F_{n(|s|)} - \tilde{\pi}(q, \cdot)\|_\infty \\
&\quad \leqq \lim_{|s|\to\infty} \|\phi^s(x) - F_{n(|s|)}\|_\infty + \overbrace{\lim_{|s|\to\infty} \|F_{n(|s|)} - \tilde{\pi}(q, \cdot)\|_\infty}^{\text{a.s. 0 by Glivenko–Cantelli}} \quad \leqq_{\text{a.s.}} \epsilon.
\end{aligned}$$

We use $n(|s|) \to \infty$ as $|s| \to \infty$, implied by the strong connectivity of our PFSA. This completes the proof. ■

— (*Statement of theorem 2.18 on derivative heap approximation*) *For s generated by a QSP, let $\mathcal{D}^s(L)$ be computed with $L = \Sigma^{O(\log(1/\epsilon))}$. If for some $x_0 \in \Sigma^{O(\log(1/\epsilon))}$, $\phi^s(x_0)$ is a vertex of the convex hull of $\mathcal{D}^s(L)$, then*

$$\text{Prob}(x_0 \textit{ is not } \epsilon\textit{-synchronizing}) \leqq e^{-|s|\epsilon O(1)}. \tag{A 9}$$

Proof. The result follows from Sanov's theorem [32] for convex set of probability distributions. Note that if $|s| \to \infty$, then x_0 is guaranteed to be ϵ-synchronizing (theorem 2.10 and corollary 2.12). Denoting the number of times we encounter x_0 in s as $n(|s|)$, and since $\mathcal{D}_\infty$ is a convex set of distributions (allowing us to drop the polynomial factor in Sanov's bound), we apply Sanov's theorem to the case of finite s:

$$\text{Prob}(\text{KL}(\phi^s(x_0) \| \wp_{x_0} \tilde{\Pi}) > \epsilon) \leqq \mathrm{e}^{-n(|s|)\epsilon}, \tag{A 10}$$

where $\text{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence [42]. From the bound [43]:

$$\tfrac{1}{4}\|\phi^s(x_0) - \wp_{x_0}\tilde{\Pi}\|_\infty^2 \leqq \text{KL}(\phi^s(x_0) \| \wp_{x_0}\tilde{\Pi}) \tag{A 11}$$

and $n(|s|) \to |s|\alpha = |s|O(1)$, where $\alpha > 0$ is the stationary probability of encountering x_0 in s, we conclude:

$$\text{Prob}(\|\phi^s(x_0) - \wp_{x_0}\tilde{\Pi}\|_\infty > \epsilon) \leqq 2\,\mathrm{e}^{-1/2|s|\epsilon O(1)} = \mathrm{e}^{-|s|\epsilon O(1)} \tag{A 12}$$

which completes the proof. ■

— (*Statement of theorem 3.1 on time complexity*) *Asymptotic time complexity of* `GenESeSS` *is*

$$\mathcal{T} = O((1/\epsilon + |Q|) \times |s| \times |\Sigma|). \tag{A 13}$$

Proof. `GenESeSS` performs the following computations.

(C1) Computation of a derivative heap by computing $\phi^s(x)$ for $O(1/\epsilon)$ strings (corollary 2.12), each of which involves reading the input s and normalization to distributions over Σ, thus contributing $O(1/\epsilon \times |s| \times |\Sigma|)$.
(C2) Finding a vertex of the convex hull of the heap, which, at worst, involves inspecting $O(\epsilon)$ points (encoded by strings generating the heap), contributing $O(1/\epsilon \times |\Sigma|)$, where each inspection is done in $O(|\Sigma|)$ time.
(C3) Finding δ, involving computing derivatives at string-identifiers (step 2), thus contributing $O(|Q| \times |s| \times |\Sigma|)$. Strong connectivity [34] requires $O(|Q| + |\Sigma|)$, and hence is unimportant.

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

(C4) Identification of arc probabilities using traversal counts and normalization, done in time linear in the number of arcs, i.e. $O(|Q| \times |\Sigma|)$.

Summing the contributions, and using $|s| > |\Sigma|$,

$$\begin{aligned}\mathcal{T} &= O(1/\epsilon \times |s| \times |\Sigma| + 1/\epsilon \times |\Sigma| + |Q| \times |s| \times |\Sigma| + |Q| \times |\Sigma|) \\ &= O((1/\epsilon + |Q|) \times |s| \times |\Sigma|)\end{aligned}$$

which completes the proof. ■

References

1. Angluin D. 1987 Queries and concept learning. *Mach. Learn.* **2**, 319–342. (doi:10.1007/BF00116828)
2. Angluin D. 2004 Queries revisited. *Theoret. Comput. Sci.* **313**, 175–194. (doi:10.1016/j.tcs.2003.11.004)
3. Ljung L. 1999 *System identification: theory for the user*, 2nd edn. Englewood Cliffs, NJ: Prentice Hall.
4. Bongard J, Lipson H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104**, 9943–9948. (doi:10.1073/pnas.0609476104)
5. Schmidt M, Lipson H. 2009 Distilling free-form natural laws from experimental data. *Science* **324**, 81–85. (doi:10.1126/science.1165893)
6. Hipel K, McLeod A. 1994 *Time series modelling of water resources and environmental systems*. Developments in Water Science, no. 45. Amsterdam, The Netherlands: Elsevier.
7. Box G, Jenkins G, Reinsel G. 1994 *Time series analysis: forecasting and control*. Forecasting and Control Series. Englewood Cliffs, NJ: Prentice-Hall.
8. Paul G. 1993 Approaches to abductive reasoning—an overview. *Artif. Intell. Rev.* **7**, 109–152. (doi:10.1007/BF00849080)
9. Angluin D. 1988 Identifying languages from stochastic examples. Technical report YALEU/DCS/RR-614, Yale University, New Haven, CT, USA.
10. Horning JJ. 1969 A study of grammatical inference (AAI7010465). PhD thesis, Stanford, CA, USA.
11. Vander-Mude A, Walker A. 1978 On the inference of stochastic regular grammars. *Inform. Control* **38**, 310–329. (doi:10.1016/S0019-9958(78)90106-7)
12. Shalizi CR, Shalizi KL. 2004 Blind construction of optimal nonlinear recursive predictors for discrete sequences. In *AUAI '04: Proc. 20th Conf. on Uncertainty in Artificial Intelligence*, pp. 504–511. Arlington, VA: AUAI Press.
13. Ron D, Singer Y, Tishby N. 1998 On the learnability and usage of acyclic probabilistic finite automata. *J. Comput. Syst. Sci.* **56**, 133–152. (doi:10.1006/jcss.1997.1555)
14. Carrasco RC, Oncina J. 1999 Learning deterministic regular grammars from stochastic samples in polynomial time. *ITA* **33**, 1–20.
15. Clark A, Thollard F. 2004 Pac-learnability of probabilistic deterministic finite state automata. *J. Mach. Learn. Res.* **5**, 473–497.
16. Ron D, Singer Y, Tishby N. 1994 Learning probabilistic automata with variable memory length. In *Proc. 7th Annu. Conf. on Computational Learning Theory*, pp. 35–46. New York, NY: ACM.
17. McTague CS, Crutchfield JP. 2006 Automated pattern detection: an algorithm for constructing optimally synchronizing multi-regular language filters. *Theoret. Comput. Sci.* **359**, 306–328. (doi:10.1016/j.tcs.2006.05.002)
18. Beran J. 1994 *Statistics for long-memory processes*. Monographs on Statistics and Applied Probability, no. 61. London, UK: Chapman & Hall.
19. Leland WE, Taqqu MS, Willinger W, Wilson DV. 1993 On the self-similar nature of ethernet traffic. *SIGCOMM Comput. Commun. Rev.* **23**, 183–193. (doi:10.1145/167954.166255)
20. Baillie RT. 1996 Long memory processes and fractional integration in econometrics. *J. Econometrics* **73**, 5–59. (doi:10.1016/0304-4076(95)01732-1)
21. Willinger W, Taqqu MS, Teverovsky V. 1999 Stock market prices and long-range dependence. *Finance Stochast.* **3**, 1–13. (doi:10.1007/s007800050049)

Downloaded from http://rsta.royalsocietypublishing.org/ on March 10, 2016

22. Polychronaki GE, Ktonas PY, Gatzonis S, Siatouni A, Asvestas PA, Tsekou H, Sakas D, Nikita KS. 2010 Comparison of fractal dimension estimation algorithms for epileptic seizure onset detection. *J. Neural Eng.* **7**, 046007. (doi:10.1088/1741-25601/7/4/046007)
23. Osorio I, Frei MG. 2007 Hurst parameter estimation for epileptic seizure detection. *Commun. Inf. Syst.* **7**, 167–176.
24. Montanari A, Taqqu M, Teverovsky V. 1999 Estimating long-range dependence in the presence of periodicity: an empirical study. *Math. Comput. Modell.* **29**, 217–228. (doi:10.1016/S0895-7177(99)00104-1)
25. Ritke R, Hong X, Gerla M. 2001 Contradictory relationship between Hurst parameter and queueing performance (extended version). *Telecommun. Syst.* **16**, 159–175. (doi:10.1023/A:1009063114616)
26. Hopcroft JE, Motwani R, Ullman JD. 2001 *Introduction to automata theory, languages, and computation*, 2nd edn. Reading, MA: Addison-Wesley.
27. Gavaldà R, Keller PW, Pineau J, Precup D. 2006 PAC-learning of Markov models with hidden state. In *Proc. 11th European Conf. on Machine Learning* (eds J Fürnkranz, T Scheffer, M Spiliopoulou). Lecture Notes in Computer Science, vol. 4212, pp. 150–161. Berlin, Germany: Springer.
28. Castro J, Gavaldà R. 2008 Towards feasible PAC-learning of probabilistic deterministic finite automata. In *Proc. 9th Int. Colloquium on Grammatical Inference* (eds A Clark, F Coste, L Miclet). Lecture Notes in Computer Science, vol. 5278, pp. 163–174. Berlin, Germany: Springer.
29. Valiant LG. 1984 A theory of the learnable. *Commun. ACM* **27**, 1134–1142. (doi:10.1145/1968.1972)
30. Chattopadhyay I, Ray A. 2008 Structural transformations of probabilistic finite state machines. *Int. J. Control* **81**, 820–835. (doi:10.1080/00207/70701704746)
31. Topsøe F. 1970 On the Glivenko–Cantelli theorem. *Probab. Theory Relat. Fields* **14**, 239–250. (doi:10.1007/BF01111419)
32. Csiszár I. 1984 Sanov property, generalized I-projection and a conditional limit theorem. *Ann. Probab.* **12**, 768–793. (doi:10.1214/aop/1176993227)
33. Barber CB, Dobkin DP, Huhdanpaa H. 1996 The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* **22**, 469–483. (doi:10.1145/235815.235821)
34. Tarjan R. 1972 Depth-first search and linear graph algorithms. *SIAM J. Comput.* **1**, 146–160. (doi:10.1137/0201010)
35. Angluin D. 1992 Computational learning theory: survey and selected bibliography. In *Proc. 24th Annu. ACM Symp. on Theory of Computing*, pp. 351–369. New York, NY: ACM.
36. Kearns MJ, Vazirani UV. 1994 *An introduction to computational learning theory*. Cambridge, MA: MIT Press.
37. Kearns M, Mansour Y, Ron D, Rubinfeld R, Schapire RE, Sellie L. 1994 On the learnability of discrete distributions. In *Proc. 26th Annu. ACM Symp. on Theory of Computing*, pp. 273–282. New York, NY: ACM.
38. Newton H. 1988 *Timeslab: a time series analysis laboratory*. The Wadsworth & Brooks/Cole Statistics/Probability Series. Pacific Grove, CA: Brooks/Cole.
39. Nicholson AJ. 1957 The self-adjustment of populations to change. *Cold Spring Harbor Symp. Quant. Biol.* **22**, 153–173.
40. Bogdanovic S, Imreh B, Ciric M, Petkovic T. 1999 Directable automata and their generalizations—a survey. *Novi Sad J. Math.* **29**, 31–74.
41. Ito M, Duske J. 1984 On cofinal and definite automata. *Acta Cybern.* **6**, 181–189.
42. Lehmann EL, Romano JP. 2005 *Testing statistical hypotheses*, 3rd edn. Springer Texts in Statistics. New York, NY: Springer.
43. Tsybakov A. 2009 *Introduction to nonparametric estimation*. Springer Series in Statistics. Berlin, Germany: Springer.