Signal Processing 93 (2013) 252–260

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Adaptive pattern classification for symbolic dynamic systems ☆

Yicheng Wen, Kushal Mukherjee, Asok Ray *

Pennsylvania State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 15 January 2012
Received in revised form
10 June 2012
Accepted 4 August 2012
Available online 11 August 2012

Keywords:
Statistical pattern classification
Symbolic dynamics
Probabilistic finite state automata

ABSTRACT

This paper addresses pattern classification in dynamical systems, where the underlying algorithms are formulated in the symbolic domain and the patterns are constructed from symbol strings as probabilistic finite state automata (PFSA) with (possibly) diverse algebraic structures. A combination of Dirichlet and multinomial distributions is used to model the uncertainties due to the (finite-length) string approximation of symbol sequences in both training and testing phases of pattern classification. The classifier algorithm follows the structure of a Bayes model and has been validated on a simulation test bed. The results of numerical simulation are presented for several examples.

1. Introduction

Commonly used classification algorithms may not always be well-suited for applications to nonlinear dynamical systems [1–3]. This phenomenon has led to departure from the conventional continuous-domain modeling towards a formal language-theoretic paradigm [4] in the symbolic domain [5,6], where a symbol sequence is generated from the output of the dynamical system by partitioning (also called quantization) of the time-series data. Thereafter, a probabilistic finite state automaton (PFSA) is constructed from the (finite-length) symbol string via one of the construction algorithms (e.g., [7–9]). Due to the quasi-stationarity assumption in the construction of PFSA, it may not be feasible to obtain sufficiently long strings of symbols in both training and testing phases of classification. Therefore, the estimated parameters of the resulting PFSA model may not be precise.

The problem of symbol string approximation has been addressed in the literature from different perspectives. For example, Kumar et al. [10] reported approximate string matching as a process of searching for optimal alignment of two finite-length strings in which comparable patterns may not be obvious; and Raskhodnikova et al. [11] approximated the compressibility of a symbol string with respect to a fixed compression scheme in sublinear time. In contrast, the current paper derives the probability distributions for constructing the PFSA to fit the classification algorithms via symbol string approximation. These algorithms are formulated by quantitatively incorporating the effects of (finite-length) symbol strings on both training and testing phases of pattern classification. In this setting, pertinent information (e.g., the probability morph matrix and state transition function) derived from a PFSA model serves as a feature vector even though the structures of PFSA may be dissimilar. The approach undertaken in the current paper is unique to the knowledge of the authors.

While previous publications in the above research area have addressed the important issues of alphabet size selection (e.g., [12]), partitioning of data sets for symbol generation (e.g., [13,14]), modeling of the hidden

☆ This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376, and by the U.S. Office of Naval Research under Grant No. N00014-09-1-0688. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

* Corresponding author.
E-mail addresses: yxw167@psu.edu (Y. Wen), kum162@psu.edu (K. Mukherjee), axr2@psu.edu (A. Ray).

0165-1684/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2012.08.002

structure of symbolic systems (e.g., [7–9]) and model order reduction of Markov chains [15], the goal of the current paper is to construct a Bayesian classifier for identification of the probability morph matrices of PFSA based on finite-length time series data. In this context, *a priori* and *a posteriori* models of uncertainties have been constructed by using Dirichlet and multinomial distributions, respectively. Other researchers (e.g., [16]) have used Dirichlet-compound-Multinomial (DCM) models to address a variety of applications, especially in the area of text modeling. Specifically, DCM distributions provide a concise approach to model *burstiness* [17,18], i.e., if an event occur once, it is likely to occur repeatedly in the future.

The paper is organized in four sections, including the present section, and an Appendix. In Section 2, the basic concepts and notations of semantic models are presented. In Section 3, the classification problem is defined and the adaptive classifier is constructed. Numerical results are shown in Section 4 to illustrate the effectiveness of the designed classifier by considering the van der Pol equation with different parameter values. The paper is concluded and summarized with recommendations for the future research in Section 5. The Appendix summarizes the pertinent principles of the Dirichlet distribution.

2. Preliminaries

In the formal language theory [19], an alphabet Σ is a (non-empty finite) set of symbols. A symbol string x over Σ has a finite length and the length of x, denoted by $|x|$, represents the number of symbols in x. The Kleene closure of Σ, denoted by $\Sigma^\star$, is the set of all finite-length symbol strings including the null string ϵ, where $|\epsilon|=0$. The set of all strings of length $d \in \mathbb{N}_0 \triangleq \{0,1,2,\ldots\}$ is denoted by $\Sigma^d \subsetneq \Sigma^\star$. The string xy is called the concatenation of two strings x and y.

Definition 2.1 (*PFSA*). A probabilistic finite state automaton (PFSA) is a tuple $G \triangleq (Q,\Sigma,\delta,\Pi,q_o)$, where

- The input alphabet Σ is a nonempty finite set of symbols, i.e., $|\Sigma| \in \mathbb{N}_1 \triangleq \{1,2,\ldots\}$.
- The set of states Q is nonempty and finite, i.e., $|Q| \in \mathbb{N}_1$.
- The state transition function $\delta : Q \times \Sigma \rightarrow Q$ naturally induces an extended transition function $\delta^\star : Q \times \Sigma^\star \rightarrow Q$ such that $\delta^\star(q,\epsilon)=q$ and $\delta^\star(q,\omega\sigma)=\delta(\delta^\star(q,\omega),\sigma)$ for every $q \in Q$, $\omega \in \Sigma^\star$ and $\sigma \in \Sigma$.
- The morph function $\pi : Q \times \Sigma \rightarrow [0,1]$ is an output mapping that satisfies the condition: $\sum_{\sigma \in \Sigma} \pi(q,\sigma)=1$ for all $q \in Q$. The morph function π has a matrix representation Π, called the (probability) morph matrix $\Pi_{ij} \triangleq \pi(q_i,\sigma_j), \forall q_i \in Q$ and $\forall \sigma_j \in \Sigma$. Note that Π is a $(|Q| \times |\Sigma|)$ stochastic matrix, i.e., each element of Π is non-negative and each row sum of Π is equal to 1.
- The start state $q_o \in Q$.

Remark 2.1. For a PFSA $G=(Q,\Sigma,\delta,\Pi,q_o)$, it is possible to generate a random symbol string $S \triangleq \{\sigma_j\}_{j=0}^{N-1}$, where each $\sigma_j \in \Sigma$ and $N \in \mathbb{N}_1$. Let $q_k \in Q$ denote the state of S at an instant k before the symbol σ_k is generated. Upon generation of the symbol σ_k based on the symbol distribution at state q_k, namely, $\pi(q_k,\cdot)$, the next state follows from the transition $q_{k+1}=\delta(q_k,\sigma_k)$.

Definition 2.2 (*Irreducibility*). A PFSA $G=(Q,\Sigma,\delta,\Pi,q_o)$ is called irreducible if, for any $q_i,q_j \in Q$, there exits a symbol string $\omega_{ij} \in \Sigma^\star$ such that $\delta^\star(q_i,\omega_{ij})=q_j$.

Definition 2.3 (*State transition matrix*). For every PFSA $G=(Q,\Sigma,\delta,\Pi,q_o)$, there is an associated $(|Q| \times |Q|)$ stochastic matrix P, called the state transition probability matrix, which is defined as follows:

$$P_{jk} = \sum_{\sigma:\delta(q_j,\sigma)=q_k} \pi(q_j,\sigma) \tag{1}$$

Remark 2.2. It follows from Perron–Frobenius Theorem [20] that every irreducible stochastic matrix has one and only one unity eigenvalue; each element of the left eigenvector corresponding to the unity eigenvalue is non-zero and has the same sign. Therefore, the state probability vector that is the left eigenvector corresponding to the unity eigenvalue is unique subject to the normalization with the sum of all its (strictly positive) elements being unity. Equivalently, for every $|Q| \times |Q|$ irreducible stochastic matrix P, there exists a unique $(1 \times |Q|)$ row-vector p such that

$$pP=p, \quad \text{where } p_j>0\ \forall j \quad \text{and} \quad \sum_{j=1}^{|Q|} p_j = 1 \tag{2}$$

where p represents a stationary probability distribution over the states of the PFSA G.

Definition 2.4 (*Synchronizing string in PFSA*). Let $G=(Q,\Sigma,\delta,\Pi,q_o)$ be a PFSA. Then, $\omega \in \Sigma^\star$ is called a synchronizing string of symbols for a state $q \in Q$ if $\delta^\star(q_i,\omega)=q$ for all $q_i \in Q$. A PFSA is called synchronizable if there exists a synchronizing string for a state $q \in Q$.

Remark 2.3. For a synchronizable PFSA, a synchronizing string yields perfect state localization. For example, if ω is a synchronizing string for a state $q \in Q$ in the PFSA G, then the substring $\sigma_k\sigma_{k+1}\ldots$ of a string $\sigma_1\sigma_2\ldots\omega\sigma_k\sigma_{k+1}\ldots$ shall have the initial state q, although the perfect state localization of the original string may not be possible.

3. The online classification problem

Let there be K classes of symbolic systems of interest, denoted by $C_1,C_2,\ldots,C_K$, over the same alphabet Σ and each class C_i is modeled by an ergodic (i.e., irreducible) PFSA $G^i=(Q^i,\Sigma,\delta^i,\Pi^i,q_0^i)$, where $i=1,2\ldots,K$. Note that the initial state q_0^i would not have any significance eventually for a synchronizable PFSA (see Remark 2.3).

During the training phase, a symbol string $S^i \triangleq s_1^i s_2^i \ldots s_{N_i}^i$ is generated from each class C_i. If a PFSA G^i is obtained for each class by executing one of the available PFSA construction algorithms, then their structures may not necessarily be the same. Thus, having Q^i and δ^i known for all K classes, Π^i's become the only unknowns that could be selected as the feature vectors for the purpose of classification. The distribution of the morph matrix Π^i is computed in the training phase.

In the testing phase, let another symbol string $\tilde{S}$ be generated by one of the PFSA models that are constructed in the training phase. Then, the task is to classify which class this observed symbol string $\tilde{S}$ belongs to. While the previous work [7–9] has aimed at identification of a PFSA from a given symbol string, the objective of this paper is to imbed the uncertainties due to the finite length of the symbol string in the identification algorithm that would influence the final classification decision. Fig. 1 presents an overview of the classification procedure for Class C_i in terms of the following information:

(1) Hidden structure (Q^i,δ^i) of the training symbol string S^i, generated by a standard PFSA construction algorithm from a symbol string (e.g., [7–9]).
(2) Dirichlet distribution [21] of Π^i, computed from S^i and (Q^i,δ^i).
(3) Multinomial distribution [21] to obtain $\Pr(\tilde{S}|S^i)$ from the PFSA G^i and the observed string $\tilde{S}$ of symbols.

A pertinent assumption in the training phase is that each symbol string S^i is state-synchronized, which implies that it is possible to find the initial state q_o^i for S^i and therefore localize the states in G^i for each symbol in S^i. This is a realistic assumption in the training phase, because the state synchronization is usually a part of the existing PFSA construction algorithms. In many cases, the state can be localized after the occurrence of a few symbols.

Following Remark 2.3, the training data are truncated and it is possible to use a part of the data after the point where the state has been synchronized to the structure of the corresponding PFSA. Having known the initial state q_o^i for each class C_i, a sequence of states is admitted for each string S^i in the training set. Let a sequence of states of length N^i+1, belonging to the class C_i be denoted as $\mathbb{V}^i \triangleq \nu_0^i\nu_1^i\nu_2^i\cdots\nu_{N^i}^i$ with

$$\nu_0^i \triangleq q_o^i \quad \text{and} \quad \nu_{k+1}^i=\delta(\nu_k^i,\sigma_k^i) \tag{3}$$

Therefore, each row of Π^i is treated as a random vector. Let the mth row of Π^i be denoted as Π_m^i and the nth element of the mth row as $\Pi_{mn}^i \ge 0$ and $\sum_{n=1}^{|\Sigma|}\Pi_{mn}^i=1$ for each row m. Conditioned on a given current state, the probability of the subsequent symbol is independent of the past states of the string and is time-invariant, i.e., if the same state is visited again, the probabilities of occurrence of subsequent symbols would be identical. Therefore, the *a priori* probability density function $f_{\Pi_m^i|S^i}$ of the random row-vector Π_m^i, conditioned on a symbol string S^i, follows the Dirichlet distribution [22,23] as explained in the Appendix. The pertinent results are summarized below.

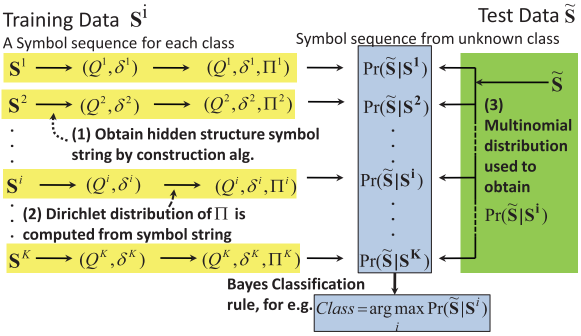
Training Data $\mathbf{S}^i$
A Symbol sequence for each class
Test Data $\tilde{\mathbf{S}}$
Symbol sequence from unknown class
$\mathbf{S}^1 \longrightarrow (Q^1,\delta^1) \longrightarrow (Q^1,\delta^1,\Pi^1)$
$\mathbf{S}^2 \longrightarrow (Q^2,\delta^2) \longrightarrow (Q^2,\delta^2,\Pi^2)$
(1) Obtain hidden structure symbol string by construction alg.
$\mathbf{S}^i \longrightarrow (Q^i,\delta^i) \longrightarrow (Q^i,\delta^i,\Pi^i)$
(2) Dirichlet distribution of Π is computed from symbol string
$\mathbf{S}^K \longrightarrow (Q^K,\delta^K) \longrightarrow (Q^K,\delta^K,\Pi^K)$
$\Pr(\tilde{\mathbf{S}}|\mathbf{S}^1)$
$\Pr(\tilde{\mathbf{S}}|\mathbf{S}^2)$
$\Pr(\tilde{\mathbf{S}}|\mathbf{S}^i)$
$\Pr(\tilde{\mathbf{S}}|\mathbf{S}^K)$
$\tilde{\mathbf{S}}$
(3) Multinomial distribution used to obtain $\Pr(\tilde{\mathbf{S}}|\mathbf{S}^i)$
Bayes Classification rule, for e.g.
$Class=\arg\max_i \Pr(\tilde{\mathbf{S}}|\mathbf{S}^i)$

Fig. 1. Overview of the classification procedure for Class C_i: (1) hidden structure (Q^i,δ^i) of the training symbol string S^i; (2) Dirichlet distribution of the morph matrix Π^i computed from S^i and (Q^i,δ^i); and (3) multinomial distribution to obtain $\Pr(\tilde{S}|S^i)$ from (Q^i,δ^i,Π^i) and the observed string $\tilde{S}$.

$$f_{\Pi_m^i|S^i}(\boldsymbol{\theta}_m^i|S^i)=\frac{1}{B(\boldsymbol{\alpha}_m^i)}\prod_{n=1}^{|\Sigma|}(\theta_{mn}^i)^{\alpha_{mn}^i-1} \tag{4}$$

where $\boldsymbol{\theta}_m^i$ is a realization of the random vector Π_m^i, namely

$$\boldsymbol{\theta}_m^i=[\theta_{m1}^i \ \ \theta_{m2}^i \ \ \ldots \ \ \theta_{m|\Sigma|}^i]$$

and the normalizing constant is

$$B(\boldsymbol{\alpha}_m^i)\triangleq\frac{\prod_{n=1}^{|\Sigma|}\Gamma(\alpha_{mn}^i)}{\Gamma(\sum_{n=1}^{|\Sigma|}\alpha_{mn}^i)} \tag{5}$$

where $\Gamma(\bullet)$ is the standard gamma function and $\boldsymbol{\alpha}_m^i=[\boldsymbol{\alpha}_{m1}^i \ \ \boldsymbol{\alpha}_{m2}^i \ \ \ldots \ \ \boldsymbol{\alpha}_{m|\Sigma|}^i]$ with

$$\alpha_{mn}^i=N_{mn}^i+1 \tag{6}$$

where N_{mn}^i is the number of times the symbol σ_n in S^i is emanated from the state q_m^i, i.e.,

$$N_{mn}^i\triangleq|\{(s_k^i,\nu_k^i):s_k^i=\sigma_n,\nu_k^i=q_m^i\}| \tag{7}$$

Recalling that s_k^i is the kth symbol in S^i, and denoting the number of occurrence of the state q_m^i in the state sequence $\mathbb{V}^i\backslash\{\nu_{N^i}^i\}$ as $N_m^i\triangleq\sum_{n=1}^{|\Sigma|}N_{mn}^i$, it follows from Eqs. (5) and (6) that

$$B(\boldsymbol{\alpha}_m^i)=\frac{\prod_{n=1}^{|\Sigma|}\Gamma(N_{mn}^i+1)}{\Gamma(\sum_{n=1}^{|\Sigma|}N_{mn}^i+|\Sigma|)}=\frac{\prod_{n=1}^{|\Sigma|}(N_{mn}^i)!}{(N_m^i+|\Sigma|-1)!} \tag{8}$$

by use of the relation $\Gamma(n)=(n-1)!\ \forall n\in\mathbb{N}_1$.

By the Markov property of the PFSA G^i, the $(1\times|\Sigma|)$ row-vectors, $\{\Pi_m^i\}, m=1,\ldots|Q|$, are statistically independent of each other. Therefore, it follows from Eqs. (4) and (8) that the *a priori* joint density $f_{\Pi^i|S^i}$ of the probability morph matrix Π^i, conditioned on the symbol string S^i, is given as

$$\begin{aligned} f_{\Pi^i|S^i}(\boldsymbol{\theta}^i|S^i) &= \prod_{m=1}^{|Q^i|} f_{\Pi_m^i|S^i}(\boldsymbol{\theta}_m^i|S^i) \\ &= \prod_{m=1}^{|Q^i|}(N_m^i+|\Sigma|-1)!\prod_{n=1}^{|\Sigma|}\frac{(\boldsymbol{\theta}_m^i)^{N_{mn}^i}}{(N_{mn}^i)!} \end{aligned} \tag{9}$$

where $\boldsymbol{\theta}^i\triangleq[(\boldsymbol{\theta}_1^i)^T \ \ (\boldsymbol{\theta}_2^i)^T \ \ \ldots \ \ (\boldsymbol{\theta}_{|Q|}^i)^T]^T\in[0,1]^{|Q|\times|\Sigma|}$.

In the testing phase, the probability of an observed symbol string $\tilde{S}$ belonging to the ith class of PFSA $(Q^i,\ \Sigma,\delta^i,\Pi^i,q_0^i)$, $i=1,\ldots,K$, is modeled by the multinomial distribution [21]. The multinomial distribution provides the probability of observing a certain distribution (e.g., frequency count) of symbols conditioned on a given state and a morph function Π^i. For an arbitrary initial state $q_0^i = q_j^i \in Q^i$, the probability of observing $\tilde{S}$ is derived below

$$\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i) = \sum_{j=1}^{|Q^i|} \Pr(q_j^i)\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i,q_j^i) \tag{10}$$

$$\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i) = \sum_{j=1}^{|Q^i|} p^i(j)\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i,q_j^i) \tag{11}$$

$$\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i) = \sum_{j=1}^{|Q^i|} \frac{N_j^i}{N^i} \prod_{m=1}^{|Q^i|} (\tilde{N}_m^i)! \prod_{n=1}^{|\Sigma|} \frac{(\Pi_{mn}^i)^{\tilde{N}_{mn}^i}}{(\tilde{N}_{mn}^i)!} \tag{12}$$

$$\Pr(\tilde{S}|Q^i,\delta^i,\Pi^i) \triangleq \Pr(\tilde{S}|\Pi^i) \quad \text{when } Q^i \text{ and } \delta^i \text{ are kept invariant} \tag{13}$$

where Eq. (10) is applied in the chain rule of conditional probability; and Eq. (11) replaces $\Pr(q^i)$ by the stationary state probability vector p^i of the identified PFSA G^i, because the operations of state transition are assumed to be statistically stationary [7] and hence, it follows that the condition $p^i(j) = \Pr(q_j^i)\ \forall j$ is satisfied by the initial state probability vector p^i; and Eq. (12) computes the stationary probability vector p^i by the (off-line) maximum likelihood estimate (MLE) for each class C_i. In addition, with the initial condition q_o specified for the observed symbol string $\tilde{S}$ and using the statistical independence among the row elements of the morph matrix, the joint distribution of the symbol string $\tilde{S}$ is obtained as the product of $|Q^i|$ multinomial distributions.

Similar to N_{mn}^i, defined earlier for S^i, let $\tilde{N}_{mn}^i$ be the number of times the symbol σ_n is emanated from the state $q_m^i \in Q^i$ in the symbol string $\tilde{S}$ in the testing phase, i.e.

$$\tilde{N}_{mn}^i \triangleq |\{\tilde{s}_k : \tilde{s}_k = \sigma_n,\ (\delta^i)^\star(q_o^i,\tilde{s}_1 \ldots \tilde{s}_{k-1}) = q_m^i\}| \tag{14}$$

where $\tilde{s}_k$ is the kth symbol in the observed string $\tilde{S}$. It is noted that $\tilde{N}_{mn}^i$ and hence $\tilde{N}_m^i \triangleq \sum_{n=1}^{|\Sigma|} \tilde{N}_{mn}^i$ cannot be computed unless the initial state corresponding to the symbol string $\tilde{S}$ is specified. Therefore, both $\tilde{N}_{mn}^i$ and $\tilde{N}_m^i$ may depend on the index j of the initial state q_j^i (see Eqs. (10)–(12)), which is not explicitly written for the conciseness of notation. Finally, the notation is simplified in Eq. (13).

The results, derived in the training phase and the testing phase, are now combined. Given a symbol string S^i in the training phase, the probability of observing a symbol string $\tilde{S}$ in the testing phase is obtained as follows

$$\begin{aligned}\Pr(\tilde{S}|S^i) &= \int \cdots \int \Pr(\tilde{S}|\Pi^i = \boldsymbol{\theta}^i) f^i_{\Pi^i|S^i}(\boldsymbol{\theta}^i|S^i)\, d\boldsymbol{\theta}^i \\ &= \int \cdots \int \sum_{j=1}^{|Q^i|} \frac{N_j^i}{N^i} \left[\prod_{m=1}^{|Q^i|} (\tilde{N}_m^i)! \prod_{n=1}^{|\Sigma|} \frac{(\theta_{mn}^i)^{\tilde{N}_{mn}^i}}{(\tilde{N}_{mn}^i)!}\right] \\ &\quad \times \prod_{m=1}^{|Q^i|} \left[(N_m^i + |\Sigma| - 1)! \prod_{n=1}^{|\Sigma|} \frac{(\theta_{mn}^i)^{N_{mn}^i}}{(N_{mn}^i)!}\, d\theta_{mn}^i\right] \\ &= \sum_{j=1}^{|Q^i|} \frac{\tilde{N}_j^i}{N^i} \prod_{m=1}^{|Q^i|} (\tilde{N}_m^i)!(N_m^i + |\Sigma| - 1)!\end{aligned}$$

$$\times \frac{\int \cdots \int \prod_{n=1}^{|\Sigma|} (\theta_{mn}^i)^{\tilde{N}_{mn}^i + N_{mn}^i}\, d\theta_{mn}^i}{\prod_{n=1}^{|\Sigma|} (\tilde{N}_{mn}^i)!(N_{mn}^i)!} \tag{15}$$

The integrand in Eq. (15) is the density function for the Dirichlet distribution up to the multiplication of a constant. Hence, it follows from Eq. (8) that

$$\begin{aligned}&\int \cdots \int \prod_{n=1}^{|\Sigma|} (\theta_{mn}^i)^{\tilde{N}_{mn}^i + N_{mn}^i}\, d(\theta_{mn}^i) \\ &\quad = \frac{\prod_{n=1}^{|\Sigma|} (\tilde{N}_{mn}^i + N_{mn}^i)!}{(\tilde{N}_m^i + N_m^i + |\Sigma| - 1)!}\end{aligned}$$

Then, it follows from Eq. (15) that

$$\begin{aligned}\Pr(\tilde{S}|S^i) &= \sum_{j=1}^{|Q^i|} \frac{N_j^i}{N^i} \prod_{m=1}^{|Q^i|} \frac{(\tilde{N}_m^i)!(N_m^i + |\Sigma| - 1)!}{(\tilde{N}_m^i + N_m^i + |\Sigma| - 1)!} \\ &\quad \times \prod_{n=1}^{|\Sigma|} \frac{(\tilde{N}_{mn}^i + N_{mn}^i)!}{(\tilde{N}_{mn}^i)!(N_{mn}^i)!} \\ &\triangleq \sum_{j=1}^{|Q^i|} \frac{N_j^i}{N^i} \cdot \Pr(\tilde{S}|S^i,q_j^i)\end{aligned} \tag{16}$$

In Eq. (16), it is recognized that $\Pr(\tilde{S}|S^i,q_j^i)$ is the probability of observing a symbol string $\tilde{S}$ provided that the symbol string S^i, belonging to the ith class, and the initial state for the observed symbol string $\tilde{S}$ are known a priori. Since, in general, the initial state may not be known a priori, it is necessary to compute $\Pr(\tilde{S}|S^i,q_j^i)$ for each possible initial state q_j^i. However, if the structure of the underlying PFSA belonging to the class C_i is synchronizable (i.e., solely dependent on Q^i and δ^i), then the initial state can be identified from the first several symbols in $\tilde{S}$ (see Remark 2.3). Under these circumstances, the initial state for the observed symbol string is assumed to be known. Consequently, the likelihood function becomes independent of the initial state, i.e., $\Pr(\tilde{S}|S^i) = \Pr(\tilde{S}|S^i,q_j^i)$.

In practice, it might be easier to compute the logarithm of $\Pr(\tilde{S}|S^i,q_j^i)$ by virtue of Stirling's approximation formula $\log(n!) \approx n\log(n) - n$ [24] because, in most cases, both N^i and $\tilde{N}$ would consist of large numbers.

The posterior probability of the observed symbol string $\tilde{S}$ belonging to the class C_i is denoted as $\Pr(C_i|\tilde{S})$ and is given as

$$\Pr(C_i|\tilde{S}) = \frac{\Pr(\tilde{S}|S^i)\,\Pr(C_i)}{\sum_{j=1}^{K} \Pr(\tilde{S}|S^j)\,\Pr(C_j)}, \quad i = 1,2,\ldots,K \tag{17}$$

where $\Pr(C_i)$ is the known prior distribution of the class C_i. Then, the classification decision is made as follows:

$$D_{class} = \arg\max_i \Pr(C_i|\tilde{S}) = \arg\max_i (\Pr(\tilde{S}|S^i)\,\Pr(C_i)) \tag{18}$$

If there is no prior information on $\Pr(C_i)$ is available, it is logical to assume a uniform distribution over the classes. In that case, the rule of classification decision becomes

$$D_{class} = \arg\max_i \Pr(\tilde{S}|S^i) \tag{19}$$

Remark 3.1. If the information on N_{mn}^i's and $\tilde{N}_{mn}^i$'s are available, no other information is needed to obtain the statistics of the symbol strings S^i's and $\tilde{S}$. Therefore, N^i_{mn}'s and $\tilde{N}^i_{mn}$'s are sufficient statistics of S^i's and $\tilde{S}$, respectively.

4. Results of numerical simulation

This section presents the results of numerical simulation, which are generated from three examples.

4.1. Example 1: single-state PFSA

Fig. 2 shows two single-state PFSA with the alphabet $\Sigma = \{1,2,3\}$, which have identical algebraic structures, but they differ in their morph probabilities; these PFSA belong to the respective classes, C_1 and C_2. Since both PSFA have only one state, each symbol in a string is independent and identically distributed. Given an observed symbol string $\tilde{S}$, the task is to identify one of the two classes to which $\tilde{S}$ belongs, i.e., to select one of the two PFSA that would more likely generate the symbol string $\tilde{S}$. In this example, two training symbol strings S^1 and S^2 are chosen, one from each class. In the testing phase, the quantities $\tilde{N}^1_{01}$, $\tilde{N}^1_{02}$, $\tilde{N}^1_{03}$ are obtained following Eq. (14), which are essentially frequency counts of the symbols 1, 2, and 3, respectively. Let η_1, η_2, and η_3 be the normalized frequency counts obtained by $\eta_k \triangleq \tilde{N}^1_{0k}/(\tilde{N}^1_{01}+\tilde{N}^1_{02}+\tilde{N}^1_{03})$, $k=1,2,3$. Two classes are generated on the simplex plane in Fig. 3 that shows how the classification boundary changes as the length of a symbol string in the testing phase is increased.

4.2. Example 2: performance comparison with the deterministic classification rule

This example presents a comparison of the proposed classification method with the deterministic classification rule [7] that does not take the length of symbol sequences into consideration. That is, the deterministic classification rule computes the morph matrices Π^i for each class i in the training phase and the morph matrix $\tilde{\Pi}$ for the observed string $\tilde{S}$ in the testing phase without considering the effects of the individual string lengths, based on the following rule:

$$D_{class} = \arg\min_i \|\tilde{\Pi} - \Pi^i\| \tag{20}$$

where all classes are assumed to have a common algebraic structure; the rationale for this assumption is that morph matrices in all classes must be of the same size. It is noted that, for classes with different algebraic structures, the PFSA in each class is first transformed to have a common structure by synchronous composition [4]. In this example, two classes with alphabet size 2 (i.e., $|\Sigma| = 2$) are considered for both classification methods, where each class is represented by a D-Markov machine [7] with depth $D=1$; therefore, they have the same algebraic structure.

The PFSA in each class is used to generate respective training symbol strings of length L_{trn}; and testing symbol strings of length L_{tst} are generated by randomly choosing a class. Then, the proposed classification method and the deterministic classification rule are evaluated by classifying the same testing strings, the classification performance is the number of correct classification counts in 1000 iterations for different values of L_{trn} and L_{tst} as listed in Table 1. It is observed, in general, that both algorithms perform better as L_{trn} and L_{tst} are made larger. Table 1 shows that the proposed method outperforms the deterministic rule; the performance gain is more evident if both L_{trn} and L_{tst} are small. That is, the proposed method is more advantageous if the available data sets are small. This is so because the uncertainty of the morph probabilities due to the finite length of both training and testing symbol strings is taken into account in the proposed classification method, but not in the deterministic algorithm.

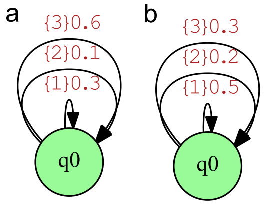

Fig. 2. Two one-state PFSA with different morph probabilities. (a) Class C_1. (b) Class C_2.

4.3. Example 3: unforced van der Pol oscillators

This section presents the classification results for a family of unforced van der Pol oscillators, whose

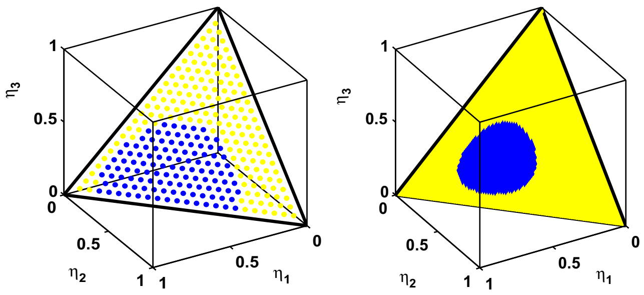

Fig. 3. Both subfigures (a) and (b) show the classification boundaries for the two classes C_1 (in light shade) and C_2 (in dark shade) for different lengths of testing string $\tilde{S}$. The training strings S^1 and S^2 for classes C_1 and C_2 are obtained from the two machines and have lengths 30 and 80, respectively. (a) Testing data length=25. (b) Testing data length=80.

Table 1
Comparison of classification performance: number of correct classifications out of 1000 iterations.

$L_{trn}\vert L_{tst}$	10	20	50	100	500
(a) The proposed classification method					
10	748	754	780	813	843
20	811	830	876	891	909
50	830	863	915	936	948
100	848	913	935	962	976
500	862	922	963	982	990
(b) The deterministic classification method					
10	747	730	773	784	783
20	758	799	854	866	876
50	814	826	884	917	934
100	816	868	911	957	965
500	845	891	940	972	979

Table 2
The values of μ for the four classes.

Class	C_1	C_2	C_3	C_4
μ	1.00	1.15	1.30	1.45

differential-equation form is given below

$$\frac{d^2x}{dt^2}+\mu(x^2-1)\frac{dx}{dt}+x=0:\mu>0$$
$$y(t)=x(t)+w(t) \tag{21}$$

where the observed output time series $y(t)$ is contaminated by additive zero-mean white Gaussian noise $w(t)$ with the signal-to-noise ratio being approximately 20 dB. A family of time series data sets are created for different values of the dissipation parameter μ in the training phase. In the testing phase, the objective is to identify the class to which the unknown parameter μ for an observed time series belongs. In this test example, let μ take four permissible values that are referred to as the four classes as shown in Table 2, which can be generalized for a larger number of classes.

Fig. 4 shows a family of plots for the output $y(t)$ belonging to the four classes of μ in the van der Pol equation, where the data for the four classes have similar characteristics. The time series data are sampled at 10 Hz to generate a discrete-time representation for each class. For the purpose of training, the length of the time series is chosen to be 8000 (i.e., a period of ~ 800 s).

The next step is to partition the data sets to yield respective symbol strings. The size of the alphabet Σ is chosen to be 15, i.e., $|\Sigma|=15$ as a trade-off between loss of information and computational complexity [12]. By adopting uniform partitioning [7], the range of the time series is divided into 15 equal intervals, each of which corresponds to a distinct symbol $\sigma_i \in \Sigma$, $i=1,2,\ldots,|\Sigma|$. The conversion to symbol strings is achieved by substituting each real-valued data point in the discrete time series by a symbol corresponding to the interval within which the data point lies. It is noted that selection of the alphabet size and the associated task of data set partitioning are critical issues in symbolic dynamic analysis; however, these issues are not highlighted in this example that focuses on explaining the relative impact of a finite data length (i.e., a (finite-length) symbol string) on the accuracy of classification decision.

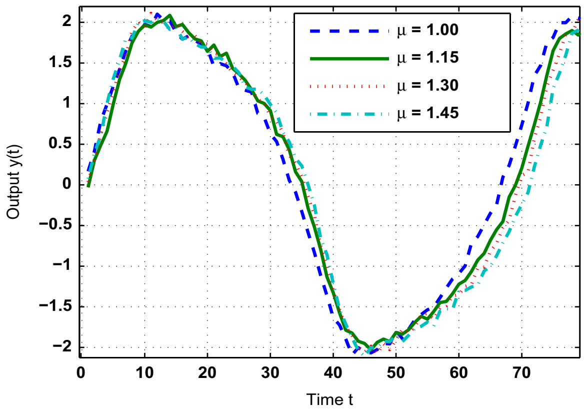

Fig. 4. Time series for four classes (van der Pol equation).

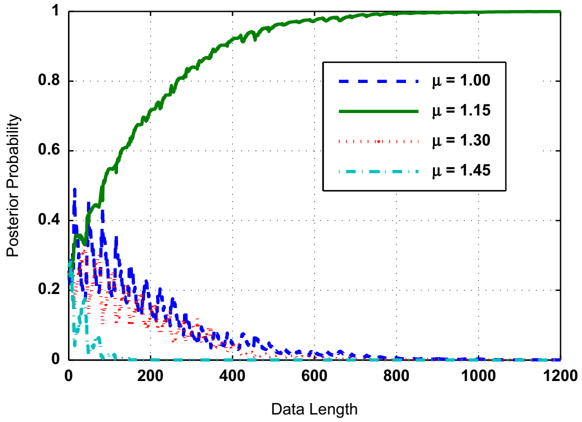

Fig. 5. Posterior probabilities for a test time series in Class C_2 ($\mu=1.15$).

The training phase commences after the symbol strings are obtained for each of the four classes in Table 2. In the D-Markov construction [7], the depth D is chosen to be 1, which implies that the probability of generation of a future symbol depends only on the last symbol, and hence the set of states is isomorphic to the symbol alphabet (i.e., $Q\equiv\Sigma$). For every class C_i, the parameters N^i_{mn} are obtained by counting the number of times the symbol σ_n is emitted from state q_m.

In the testing phase, a new time series is obtained from one of the classes and is partitioned to obtain a symbol string by using the same alphabet and partitioning as in the training phase. Following Eq. (17), the posterior probability of each class is obtained as a function of the length of the training data set. Fig. 5 shows the posterior probability of each class as a function of the length of the observed test data. It is seen in Fig. 5 that the observed string is correctly identified to belong to the class of $\mu=1.15$ as the posterior probability of the corresponding class exponentially approaches one, while each of the remaining classes (i.e., for three other values of μ) approaches zero very fast. For 600 test cases, the posterior probability of each symbol string correctly converged to either zero or one.

A classifier can also be chosen based on its receiver operating characteristic (ROC) [25]. To illustrate this concept for binary classification, let there be only two classes of van der Pol equations, namely, $\mu = 1.0$ belonging to the class C_1, and $\mu = 1.3$ belonging to the class C_2. The length of the training data for each class is chosen to be 8000. The general classification rule [25] in a symbol string $\tilde{S}$ is given by

$$\frac{\Pr(\tilde{S}|C_1)}{\Pr(\tilde{S}|C_2)} \underset{C_2}{\overset{C_1}{\gtrless}} \lambda \tag{22}$$

where the threshold λ is varied to yield the ROC curve. For the binary classification problem at hand, the ROC curve provides the trade-off between the probability of detection $P_D = \Pr\{\text{decide } C_1 | C_1 \text{ is true}\}$ and the false alarm rate $P_F = \Pr\{\text{decide } C_1 | C_2 \text{ is true}\}$. Fig. 6 exhibits a family of ROC curves for the proposed classification algorithm with varying lengths of test data. It is observed that the ROC curve improves (i.e., moves toward the top left corner) considerably as the test data length is increased from $N_{test} = 100$ to $N_{test} = 700$. Based on a family of such ROC curves, it is possible to select a best combination of P_D and N_{test} for a given P_F, which would lead to a choice of the parameter λ.

The classification capability of the proposed algorithm is tested on two classes of the van der Pol equation, C_1 and C_2, where the nominal class C_1 has its parameter $\mu = 1$ and the model parameter μ of the other class C_2 is perturbed from 1 within the range from 0.8 to 1.2. A classifier is built based on the model in Eq. (22) by choosing $\lambda = 1$. Fig. 7 depicts the classification error between the classes C_1 and C_2 obtained from 400 samples for each of the two classes. As expected, if the model parameter μ is identical for both classes C_1 and C_2, then the classification rate is approximately 50%. As μ for class C_2 deviates from 1, the error rate decreases. It is also noted that none of the three plots in Fig. 7 are symmetric about $\mu = 1$ because of the nonlinearity of the van der Pol equation.

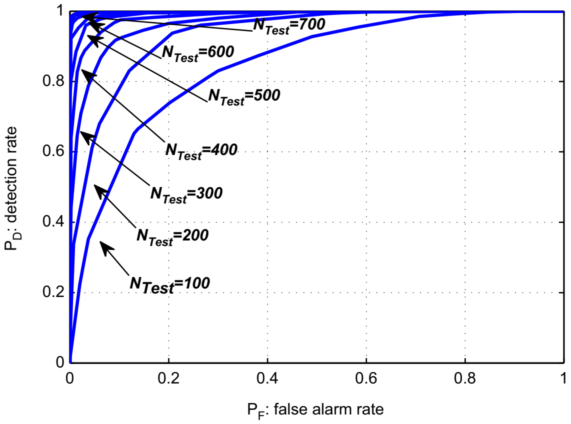

Fig. 6. ROC curves for classification with different test data lengths N_{Test}.

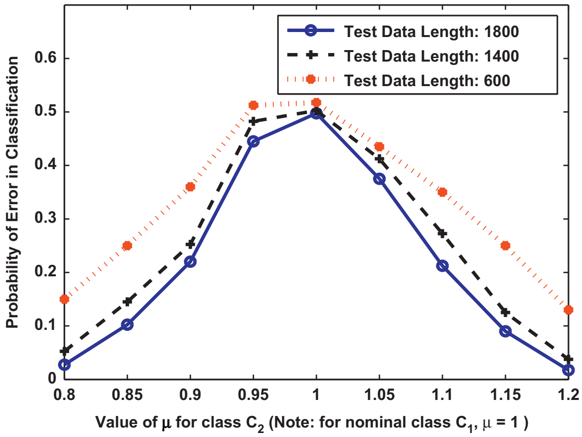

Fig. 7. The (two class) classification error as a function of testing data length. The nominal value of μ for class C_1 is 1, while the value of μ for class C_2 is shown along the x-axis.

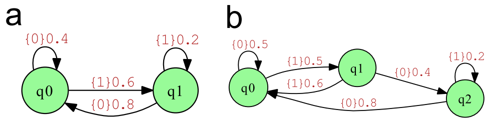

Fig. 8. PFSA of dissimilar structures. (a) C_1: D-Markov machine. (b) C_2: Non-D-Markov machine.

4.4. Example 4: PFSA of dissimilar algebraic structure

The algorithm of classifier design remains effective even if the PFSA models, belonging to different classes, have dissimilar algebraic structures, but they must be constructed over the same alphabet Σ. As an example, Fig. 8 shows two PFSA models over the alphabet $\Sigma = \{0,1\}$, where the first class (C_1) is described by a D-Markov machine [7] of depth $D=1$ and hence has two states. The second class (C_2) is described by a general synchronizable machine [9] (that is not a D-Markov machine) and similar morph probabilities are chosen for these two PFSA models. Note that although both machines in Fig. 8 are synchronizable PFSA, the D-Markov machine on the left is a subclass of shift of finite type [26] whereas the non-D-Markov machine on the right is not restricted to have that algebraic structure.

Symbol strings S^i, $i \in \{1,2\}$, of length 8000 are simulated for each of the two class with a different initial state, respectively. Another symbol string S is generated from the PFSA belonging to the class C_2 as the test case, and the maximum likelihood estimation (MLE) of the stationary probability [25] is computed based on Eq. (12). Finally, following Eq. (19), a classifier is constructed to compute the posterior probabilities of the symbol string $\tilde{S}$ for different lengths $|\tilde{S}|$, as shown in Fig. 9. Although the classifier is initially unable to make correct decisions due to indistinguishable morph probabilities for C_1 and C_2, the posterior probability of the correct class (C_2) approaches to one very fast when the observed data length becomes sufficiently large (e.g., 150) as seen in Fig. 9.

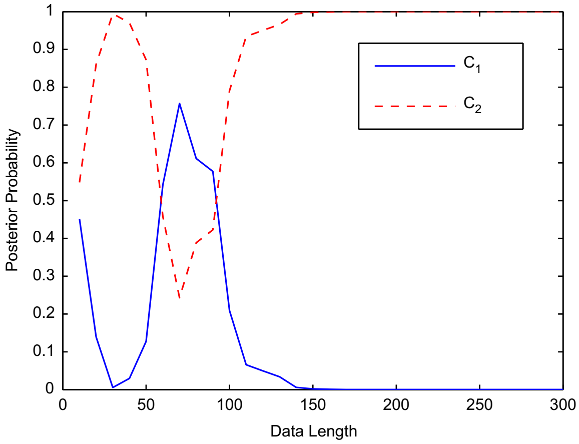

Fig. 9. Posterior probability versus data length of observed time series.

5. Summary and future work

This paper addresses pattern classification in the context of symbolic dynamic filtering (SDF) [7,27], especially if the lengths of data sets in the test phase and training phase are not equal. The algorithm is capable of handling different cases, where the patterns of interest are represented by probabilistic finite state automata (PFSA) that could have dissimilar algebraic structures for each class. In this method, the Dirichlet distribution and the multinomial distribution are used to model the uncertainties resulting from the finite length of symbol strings in both the training and testing phases. The Bayes risk function has been used for classifier design so that the results could be generated in real time as the test data are generated. The classification process can be stopped much earlier than expected if the posterior probability of one of the classes is sufficiently high. In this context, future research is recommended in the following two directions.

- *Identification of non-synchronizable classes*: Since reliability of estimation of the stationary probability vector has not been taken into account in the present formulation, the lack of training data may result in an inaccurate estimation of the a posteriori probability in Eq. (16). Therefore, the classification algorithms should focus on making uncorrupted decisions in the presence of non-synchronizable classes.
- *Sequential testing for classification decisions*: Since the objective is to make correct decisions as early as possible, the classifier should be designed in the framework of sequential testing and the resulting ROC curve needs to be investigated for performance enhancement.

Appendix A. Dirichlet distributions

In this appendix the estimated distribution of a finite set of independent and identically distributed (iid) discrete random variables is shown to follow the Dirichlet distribution [21]. Let X be a finite set of iid discrete random variables defined as

$$X=\begin{cases} 1 & \text{with probability } p_1 \\ 2 & \text{with probability } p_2 \\ \vdots & \vdots \\ N & \text{with probability } p_N \end{cases} \tag{23}$$

where $p_i \geq 0 \quad \forall i$ and $\sum_{i=1}^{N} p_i = 1$. The numerical values of p_i are estimated by repeated observation of the random variable X. Assuming that the distribution of p_i's is Dirichlet, the density function has the following structure:

$$f(p_1,p_2,\ldots,p_N)=\frac{1}{B(\alpha_1,\alpha_2,\ldots,\alpha_N)}p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_N^{\alpha_N-1}$$

$$\text{such that } p_i \geq 0 \ \forall i \text{ and } \sum_{i=1}^{N} p_i = 1 \tag{24}$$

where α_i's are the parameters of the Dirichlet distribution. $B(\alpha_1,\alpha_2,\ldots,\alpha_N)$ is the normalizing constant that is evaluated as

$$B(\alpha_1,\alpha_2,\ldots,\alpha_N)=\int_{\substack{\sum_{i=1}^{N} p_i = 1 \\ p_i \geq 0 \forall i}} p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_N^{\alpha_N-1}\,\mathrm{d}p_1\,\mathrm{d}p_2\ldots\mathrm{d}p_N$$

$$=\frac{\alpha_1!\alpha_2!\ldots\alpha_N!}{(\alpha_1+\alpha_2+\cdots+\alpha_N)!} \tag{25}$$

Let a new observation of the random variable be $X=K$. The posterior estimate of the distributions of p_i's is given by the Bayes rule [21] as

$$f(p_1,p_2,\ldots,p_N|X=K)=\frac{Prob(X=K|p_1,p_2,\ldots,p_N)f(p_1,p_2,\ldots,p_N)}{\int_{\substack{\sum_{i=1}^{N} p_i = 1 \\ p_i \geq 0 \forall i}} Prob(X=K|p_1,p_2,\ldots,p_N)f(p_1,p_2,\ldots,p_N)} \tag{26}$$

$$f(p_1,p_2,\ldots,p_N|X=K)=\frac{p_K\frac{1}{B(\alpha_1,\alpha_2,\ldots,\alpha_N)}p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_K^{\alpha_K-1}\ldots p_N^{\alpha_N-1}}{\int_{\substack{\sum_{i=1}^{N} p_i = 1 \\ p_i \geq 0 \forall i}} p_K\frac{1}{B(\alpha_1,\alpha_2,\ldots,\alpha_N)}p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_K^{\alpha_K-1}\ldots p_N^{\alpha_N-1}} \tag{27}$$

$$f(p_1,p_2,\ldots,p_N|X=K)=\frac{p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_K^{\alpha_K}\ldots p_N^{\alpha_N-1}}{\int_{\substack{\sum_{i=1}^{N} p_i = 1 \\ p_i \geq 0 \forall i}} p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_K^{\alpha_K}\ldots p_N^{\alpha_N-1}} \tag{28}$$

$$f(p_1,p_2,\ldots,p_N|X=K)=\frac{1}{B(\alpha_1,\alpha_2,\ldots,\alpha_K+1,\ldots,\alpha_N)}p_1^{\alpha_1-1}p_2^{\alpha_2-1}\ldots p_K^{\alpha_K}\ldots p_N^{\alpha_N-1} \tag{29}$$

Therefore, as the prior estimate of p_i's has a Dirichlet distribution, the posterior distribution (which is also a Dirichlet) upon observing $X=K$ is obtained by adding 1 to the corresponding parameter α_K. This process is repeated for sequentially observed instances of X.

At the initial stage, if no instance of X is observed, the least biased estimate of the distribution of p_i's is the uniform distribution on the simplex plane. Since the uniform distribution is Dirichlet with the parameters set as

$$\alpha_1=\alpha_2=\cdots=\alpha_N=1$$

it follows that the estimated distribution of p_i's is Dirichlet under the assumption of a uniform prior.

References

[1] L. Ljung, System Identification: Theory for the User, 2nd ed. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[2] O. Nelles, Nonlinear System Identification from Classical to Neural Networks and Fuzzy Models, Springer, Berlin, Germany, 2001.

[3] H. Garnier, L. Wang, Identification of continuous-time models from sampled data from classical to neural networks and fuzzy models, Springer, London, UK, 2008.

[4] I. Chattopadhyay, A. Ray, Structural transformations of probabilistic finite state machines, International Journal of Control 81 (5) (2008) 820–835.

[5] G. Pola, P. Tabuada, Symbolic models for nonlinear control systems: alternating approximate bisimulations, SIAM Journal of Control and Optimization 48 (2) (2009) 719–733.

[6] M. Vidyasagar, The complete realization problem for hidden Markov models: a survey and some new results, Mathematics of Control, Signals, and Systems 23 (December) (2011) 1–65.

[7] A. Ray, Symbolic dynamic analysis of complex systems for anomaly detection, Signal Processing 84 (7) (2004) 1115–1130.

[8] C. Shalizi, K. Shalizi, Blind construction of optimal nonlinear recursive predictors for discrete sequences, in: AUAI '04: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, Virginia, United States, 2004, pp. 504–511.

[9] I. Chattopadhyay, Y. Wen, A. Ray, S. Phoha, Unsupervised inductive learning in symbolic sequences via recursive identification of self-similar semantics, in: Proceedings of American Control Conference, San Francisco, CA, USA, June–July 2011, pp. 125–130.

[10] N. Kumar, V. Bibhu, M. Islam, S. Bhardwaj, Approximate string matching algorithm, International Journal on Computer Science and Engineering (IJSE) 2 (3) (2010) 641–644.

[11] S. Raskhodnikova, D. Ron, R. Rubinfeld, A. Shpilka, A. Smith, Sublinear algorithms for approximating string compressibility and the distribution support size, in: Electronic Colloquium on Computational Complexity, Report No. TR05-125, 2005.

[12] V. Rajagopalan, A. Ray, Symbolic time series analysis via wavelet-based partitioning, Signal Processing 86 (11) (2006) 3309–3320.

[13] M. Buhl, M. Kennel, Statistically relaxing to generating partitions for observed time-series data, Physical Review E 71 (4) (2005) 046213.

[14] S. Sarkar, K. Mukherjee, X. Jin, D. Singh, A. Ray, Optimization of symbolic feature extraction for pattern classification, Signal Processing 92 (3) (2012) 625–635.

[15] K. Deng, P. Mehta, S. Meyn, Optimal Kullback–Leibler aggregation via spectral theory of Markov chains, IEEE Transactions on Automatic Control 71 (12) (2011) 2793–2808.

[16] C. Bonafede, P. Cerchiello, A Study on Text Modelling via Dirichlet Compound Multinomial, Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin, Heidelberg, 2011.

[17] R. Madsen, D. Kauchak, C. Elkan, Modeling word burstiness using the Dirichlet distribution, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, ACM, New York, NY, USA, 2005, pp. 545–552.

[18] C. Elkan, Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution, in: Proceedings of the 23rd International Conference on Machine Learning, ICML '06, ACM, New York, NY, USA, 2006, pp. 289–296.

[19] J. Hopcroft, R. Motwani, J. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd ed. Addison-Wesley, 2001.

[20] A. Berman, R. Plemmons, Nonnegative Matrices in the Mathematical Sciences, SIAM, Philadelphia, PA, USA, 1994.

[21] S. Wilks, Mathematical Statistics, John Wiley, New York, NY, USA, 1963.

[22] T. Ferguson, A Bayesian analysis of some nonparametric problems, The Annals of Statistics 1 (2) (1973) 209–230.

[23] J. Sethuraman, A constructive definition of Dirichlet priors, Statistica Sinica 4 (1994) 639–650.

[24] R. Pathria, Statistical Mechanics, 2nd ed. Butterworth-Heinemann, Oxford, UK, 1996.

[25] V. Poor, An Introduction to Signal Detection and Estimation, 2nd ed. Springer-Verlag, New York, NY, USA, 1988.

[26] D. Lind, B. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, 1995.

[27] X. Jin, S. Gupta, K. Mukherjee, A. Ray, Wavelet-based feature extraction using probabilistic finite state automata for pattern classification, Pattern Recognition 44 (2011) 1343–1356.