*For review for presentation in **ACC '15**, Chicago, IL, USA*

A Symbolic Dynamic Filtering Approach to Unsupervised Hierarchical Feature Extraction from Time-Series Data

Adedotun Akintayo akintayo@iastate.edu
Soumik Sarkar soumiks@iastate.edu

Department of Mechanical Engineering
The Iowa State University
Ames, IA 50011, USA

Keywords: *Symbolic Dynamics; Hierarchical Feature Extraction; Time-series Analysis*

***Abstract*— This paper presents a hierarchical feature extraction technique for non-stationary time-series data that is considered to be a slow-time scale mixture of time-series segments which are quasi-stationary at a faster time-scale. The problem is to model an unknown number of unique stationary segments at the low level while capturing their switching characteristics at a higher level. Symbolic Dynamic Filtering (SDF) has been recently reported in literature as a tool for extracting spatiotemporal features from stationary time-series data. It has been shown to very efficient for early detection of anomalies (i.e., deviations from the nominal behavior) in complex dynamical systems. This paper extends the concept to develop an online (i.e., using streaming data) method that can handle quasi-stationary data to model both low and high level characteristics as Probabilistic Finite State Automata (PFSA) in an unsupervised manner (i.e., without knowing the number of unique stationary characteristics present at the low level). The algorithm is evaluated on simulated time series data generated from a nonlinear active electronic system based on the chaotic Duffing equation.**

1. Introduction

The problem of hierarchical feature extraction appear as a key problem in many application areas such as robotics, complex system modeling and image processing. For autonomous perception issues in robotics applications, environmental features are extracted in a hierarchical manner where lower level features may signify objects in the scene and higher level features represent contextual information needed for planning [1]. Similarly, hierarchical feature extraction in complex systems falls under the category of switched and hybrid system modeling approaches [2]. Recent success of deep learning in image, video and speech processing applications shows the efficacy of hierarchical feature extraction as a machine learning approach [3], [4]. One of the key innovations that came out of the deep learning community is learning hierarchical features in an unsupervised manner with deep Boltzmann machines [5]. Similar problems are being investigated in the nonparametric modeling community as well. For example, hierarchical Dirichlet process over hidden Markov models (HDP-HMM) have been shown to be efficient for automatic speaker diarisation problems [6], where 'who spoke when' [7] have to be identified from a audio time-series without knowing how many speakers there are. Cognitive processes in humans also show that ideas are generated in an adaptive and hierarchical manner. This conjecture has seen a lot of interest and success in modeling human learning and reasoning processes using probabilistic programming concepts [8].

In the context of time-series feature extraction, recently developed symbolic dynamic filtering (SDF) [9], [10] has been shown to be an efficient tool for data-driven modeling of dynamical systems. This nonlinear feature extraction tool has been shown to yield superior performance in terms of early detection of anomalies and robustness to measurement noise in comparison to other techniques such as Principal Component Analysis (PCA), Neural Networks (NN) and Bayesian filtering techniques [11]. Successful applications of SDF includes a variety of complex systems such as nuclear power plants [12], coal-gasification systems [13], ship-board auxilliary systems [14] and gas turbine engines [15], [16]. One of the primary advantage of this method is memory and computational efficiency as it provides a low-dimensional representation of the underlying dynamical system using the time-series data of the observable variables [17]. It does not involve a space of hidden (latent) variables as in HMM type models. Therefore, it can be used for on-line real-time learning and adaptation which may be an issue for deep learning and nonparametric techniques. However, SDF approximates of a symbolic time-series as a Markov chain of certain order

(in the form of a Probabilistic Finite State Automaton (PFSA)) and therefore involves an assumption that the time-series is statistically stationary [9]. This assumption limits SDF to model non-stationary time-series data that can be considered as a slow-time scale mixture of time-series segments which may be quasi-stationary at a faster time-scale. But SDF can model each unique characteristics present in the time-series as one PFSA. The entire time-series in that case can be expressed as a higher-level PFSA whose states are the automata obtained for different unique characteristics. This paper proposes a novel algorithm to learn such a model using streaming data in an unsupervised manner (i.e., without knowing how many unique characteristics or classes are present in data). Note, such modeling architecture is notionally similar to that of switched linear dynamical systems (SLDS) [18]. However, SDF is inherently a nonlinear approach and therefore, the learnt model in this case can be considered as a switched nonlinear dynamical system. To validate the efficacy of the proposed approach, a nonlinear active electronic system based on the chaotic Duffing equation [19] is chosen to be the underlying dynamical system generating the time-series data.

The paper is organized in five sections including the present one. Section 2 presents a brief background of the SDF framework along with other statistical tools used in the proposed algorithm. The on-line SDF-based hierarchical feature extraction formulation and algorithm are presented in Section 3. Section 4 provides validation results and discussions based experiments on the chaotic Duffing system. Finally, the paper is summarized and concluded in Section 5 with recommendations of future work.

2. Background and Motivation

Extraction of statistical features from time-series data generated from a dynamical system can be posed as a two time-scale problem. The fast time-scale is related to response time of the process dynamics. Let us assume that over a window of data acquisition, the dynamic behavior of the system remains invariant, i.e., the process is quasi-stationary at the fast time-scale. Such a fast time-scale window can be called an epoch in a slow time-scale. The slow time-scale is related to the time span over which deviations (e.g., parametric changes) may occur and the system may exhibit non-stationary dynamics. The original formulation of SDF involves modeling a single quasi-stationary characteristics with a Probabilistic Finite State Automaton (PFSA). However, a general dynamical system typically produces non-stationary data (by switching among different quasi-stationary behaviors) due to change in operating point or parametric condition over the slow time-scale. Therefore, the goal here is to automatically capture different quasi-stationary characteristics with different PFSA. At a higher logical level, each of such PFSA can act as states and a higher level PFSA can capture transition characteristics among those states as shown in Fig. 1. Thus, a SDF based non-stationary time-series feature extraction problem is posed as learning a hierarchical PFSA representation. The primary technical challenge is to learn such a model in an unsupervised manner, i.e., without knowing the number of unique stationary characteristics present in the data (or, the number of states needed for the higher-level PFSA).

While details of the original SDF formulation can be found in [9], [16], a brief review is presented in the sequel for completeness. The section also discusses the concepts of the Chinese Restaurant Process (CRP) and stickiness factor that are used in the proposed formulation.

A. Mathematical Formulation of SDF

The first step of SDF is an abstraction process that symbolizes the continuous space time-series data obtained from a dynamical systems. In Symbolic Dynamics literature, this quantization process is known as partitioning [9]. There are many ways of partitioning reported in the literature [20], [21] depending on different objective functions. However, the focus of this paper is modeling a symbol sequence (using PFSA) obtained after partitioning.

A PFSA is a 4-tuple $G \triangleq (Q, \Sigma, \delta, \Pi)$. The alphabet Σ is a nonempty finite set of symbols. The set of states Q is nonempty and finite. As a simplifying assumption, this paper considers only a class of PFSA, known as D-Markov machines [9]. In D-Markov machines, the states are strings of the past D symbols, where the positive integer D is is called the depth of the machine and the number of states $|Q| \leq |\Sigma|^D$. The state transition function $\delta : Q \times \Sigma \rightarrow Q$ indicates the new state given the previous state and an observed symbol. In addition, 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, where $\Pi_{ij} \triangleq \pi(q_i, \sigma_j), \forall q_i \in Q$ and $\forall \sigma_j \in \Sigma$. Note that Π is a $(|Q| \times |\Sigma|)$ matrix where each element of Π is non-negative and each row sum of Π is equal to 1. Due to the assumption of quasi-stationarity of the observed sensor data, the PFSA $G \triangleq (Q, \Sigma, \delta, \Pi)$ is not dependent on the initial state $q_0 \in Q$.

With this setup, Π acts a low-dimensional representation of the original quasi-stationary time-series which can be learnt using simple frequency counting from a set of training data. At the testing phase, if data originates from a different (possibly anomalous) condition of the system, Π (computed in the same manner) will

be significantly different from the training stage. Thus SDF can be used to detect changes in the underlying dynamical system.

B. Chinese Restaurant Process and Stickiness Factor

This subsection briefly describes a couple of basic statistical concepts used in the proposed formulation, namely the Chinese Restaurant Process (CRP) and the stickiness factor. Recently, these ideas have been extensively used in nonparametric modeling and therefore details can be found in the related literature [22], [6].

CRP is an induced distribution over partitions or clusters which is based on De Finetti's theorem [22]. The illustrative example given for CRP (the reason behind its name) involves a fictitious Chinese restaurant with potentially an infinite number of tables [23]. Given this setup the discrete time stochastic process is described by a probability distribution that determines the table assignment of a new ($k+1^{th}$) customer. The new customer can choose an already occupied (by previous k customer(s)) table o within the set of occupied tables O with a probability

$$Pr_\gamma(o \in O) = \frac{\mathbb{C}(o)}{[\sum_{x \in O} \mathbb{C}(x)] + \gamma} \tag{1}$$

where, $\mathbb{C}(\cdot)$ denotes a concentration or strength function. Or, the new customer can choose a new (previously unoccupied) table with a probability

$$Pr_\gamma(o_{new}) = \frac{\gamma}{[\sum_{x \in O} \mathbb{C}(x)] + \gamma} \tag{2}$$

Note that this is one simple definition of CRP among many variations available in literature. This paper uses the definition mentioned above to decide whether a new data segment should be modeled with an existing PFSA or a new PFSA should be created.

While induction of CRP can help in deciding the need for a new PFSA model, noise and spurious disturbance present in real data can drive the decision system unstable. That is many unnecessary new PFSA may get generated and the decision may then fluctuate between different PFSA that are close to each other based on an appropriate metric. Similar situation arises in other unsupervised techniques as well such as HDP-HMM. To prevent this scenario, [6] included a stickiness factor in the formulation as counter measure while assigning class (or cluster) to a new data point. The basic idea is to introduce a positive bias on the class assigned to the previous data point. In the present study, the overall time-series is composed of segments of quasi-stationary time-series data that can span over many slow time epochs. Therefore, this is a realistic assumption as a new slow time epoch data most likely has the same quasi-stationary characteristics as the previous slow time epoch data. This also aligns with the fact that typically a real system remains in a certain operating point or parametric condition for some time before switching to a new one.

3. METHODOLOGY

The section describes the proposed algorithm to build a two tier PFSA model for a streaming non-stationary time-series data as shown in Fig. 1. At the lower tier,

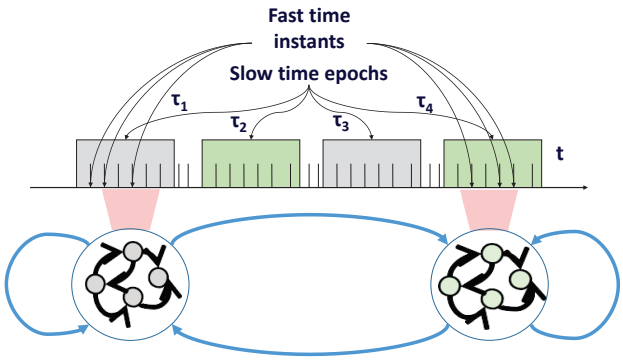

Fig. 1. Schematics of Hierarchical PFSA based Feature Extraction

the goal is to create one PFSA model for each unique quasi-stationary characteristics present in the data. Those PFSAs act as states for the second tier and another PFSA is identified to capture the transition of the system among those states. Therefore the second tier PFSA can be thought of as PFSA of PFSAs. The unsupervised online algorithm is initialized with learning a PFSA with data from the first slow time epoch. Let that PFSA class be denoted by C_1. Now, from the second slow time epoch onwards, the problem becomes identifying whether a new epoch belongs to C_1 or there should be a new class representing that data. Therefore, the affinity of a new slow time epoch to C_1 needs to be quantified. In general, when there are more classes of PFSA present, this problem can be formulated as a classification problem that computes the probability of a new slow time epoch belonging to a certain class. However, a typical slow time epoch may not have sufficient data points to estimate a PFSA Π matrix as discussed in Section 2. Therefore, an inference algorithm is required that can perform classification using relatively small length of data. Such an algorithm was developed recently in [24] and briefly outlined in Section 3-A for completeness. Once the probabilities of a new slow time epoch belonging to different existing classes are obtained, a decision is made to either assign the slow time epoch to an existing class or to create a new class. This process uses CRP and the stickiness factor which as described in Section 3-B. Once a slow time epoch is assigned to a class C_i, the data of that epoch gets used

as a part of the training data for that class from the next iteration.

A. Online Classification of a Slow Time Epoch

Let there be K classes of quasi-stationary characteristics already identified in the data. They are denoted by $C_1, C_2, \ldots, C_K$, over the same alphabet Σ and each class C_i is modeled by an ergodic (equivalently, irreducible) PFSA $G^i = (Q^i, \Sigma, \delta^i, \Pi^i)$, where $i = 1, 2 \ldots, K$. Also for each class C_i, let a symbol string $S^i \triangleq s_1^i s_2^i \ldots s_{N_i}^i$ be already identified from the streaming data. The state transition function δ and the set of states Q of the D-Markov machine are fixed by choosing an appropriate depth D and let the (probability) morph matrix be denoted by Π^i. To perform inference with small length of data, each row of Π^i is treated as a random vector. Let the m^{th} row of Π^i be denoted as Π_m^i and the n^{th} element of the m^{th} row as $\Pi_{mn}^i \geq 0$ and $\sum_{n=1}^{|\Sigma|} \Pi_{mn}^i = 1$. 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 [25] [26] as described below.

$$f_{\Pi_m^i|S^i}(\boldsymbol{\theta}_m^i|S^i) = \frac{1}{B(\boldsymbol{\alpha}_m^i)} \prod_{n=1}^{|\Sigma|} (\theta_{mn}^i)^{\alpha_{mn}^i - 1} \tag{3}$$

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

$$\boldsymbol{\theta}_m^i = \begin{bmatrix} \theta_{m1}^i & \theta_{m2}^i & \cdots & \theta_{m|\Sigma|}^i \end{bmatrix}$$

and the normalizing constant is

$$B(\boldsymbol{\alpha}_m^i) \triangleq \frac{\prod_{n=1}^{|\Sigma|} \Gamma(\alpha_{mn}^i)}{\Gamma(\sum_{n=1}^{|\Sigma|} \alpha_{mn}^i)} \tag{4}$$

where $\Gamma(\bullet)$ is the standard gamma function, and $\boldsymbol{\alpha}_m^i = \begin{bmatrix} \boldsymbol{\alpha}_{m1}^i & \boldsymbol{\alpha}_{m2}^i & \cdots & \boldsymbol{\alpha}_{m|\Sigma|}^i \end{bmatrix}$ with

$$\alpha_{mn}^i = N_{mn}^i + 1 \tag{5}$$

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

$$N_{mn}^i \triangleq \left| \{(s_k^i, v_k^i) : s_k^i = \sigma_n, v_k^i = q_m\} \right| \tag{6}$$

where s_k^i is the k^{th} symbol in S^i and v_k^i is the k^{th} state as derived from the symbolic sequence S^i. Recall that a state is defined as a string of D past symbols. The the number of occurrence of the state q_m in the state sequence is given by $N_m^i \triangleq \sum_{n=1}^{|\Sigma|} N_{mn}^i$. It follows from Eqs. (4) and (5) that

$$B(\boldsymbol{\alpha}_m^i) = \frac{\prod_{n=1}^{|\Sigma|} \Gamma(N_{mn}^i + 1)}{\Gamma(\sum_{n=1}^{|\Sigma|} N_{mn}^i + |\Sigma|)} = \frac{\prod_{n=1}^{|\Sigma|} (N_{mn}^i)!}{(N_m^i + |\Sigma| - 1)!} \tag{7}$$

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. (17) and (7) 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

$$\begin{aligned} f_{\Pi^i|S^i}(\boldsymbol{\theta}^i|S^i) &= \prod_{m=1}^{|Q|} f_{\Pi_m^i|S^i}\left(\boldsymbol{\theta}_m^i|S^i\right) \\ &= \prod_{m=1}^{|Q|} \left(N_m^i + |\Sigma| - 1\right)! \prod_{n=1}^{|\Sigma|} \frac{(\boldsymbol{\theta}_m^i)^{N_{mn}^i}}{(N_{mn}^i)!} \end{aligned} \tag{8}$$

where $\boldsymbol{\theta}^i = \begin{bmatrix} (\boldsymbol{\theta}_1^i)^T & (\boldsymbol{\theta}_2^i)^T & \cdots & (\boldsymbol{\theta}_{|q|}^i)^T \end{bmatrix} \in [0,1]^{|Q| \times |\Sigma|}$

With this setup, let a new slow time epoch contains a symbol string $\widetilde{S}$. Now, the probability that the symbol string belonging to a particular class of PFSA, $(Q, \Sigma, \delta, \Pi^i)$ is a product of independent multinomial distribution [27] given that the exact morph matrix Π^i is known.

$$\begin{aligned} &\Pr\left(\widetilde{S}|Q, \delta, \Pi^i\right) \\ &= \prod_{m=1}^{|Q|} (\widetilde{N}_m)! \prod_{n=1}^{|\Sigma|} \frac{\left(\Pi_{mn}^i\right)^{\widetilde{N}_{mn}}}{(\widetilde{N}_{mn})!} \qquad (9) \\ &\triangleq \Pr\left(\widetilde{S}|\Pi^i\right) \quad \text{as } Q \text{ and } \delta \text{ are kept invariant} \qquad (10) \end{aligned}$$

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

$$\widetilde{N}_{mn} \triangleq \left| \{(\tilde{s}_k, \tilde{v}_k) : \tilde{s}_k = \sigma_n, \tilde{v}_k = q_m\} \right| \tag{11}$$

where $\tilde{s}_k$ is the k-th symbol in the observed string $\widetilde{S}$ and $\tilde{v}_k$ is the k-th state derived from $\widetilde{S}$. It is noted that $\widetilde{N}_m \triangleq \sum_{n=1}^{|\Sigma|} \widetilde{N}_{mn}$.

Now, equations 8 and 9 can be combined to obtain the probability of a symbol string $\widetilde{S}$ belonging to a class characterized by already observed symbol string S^i. With the derivation presented in [24], the following expression can be obtained for the probability.

$$\begin{aligned} \Pr(\widetilde{S}|S^i) = &\prod_{m=1}^{|Q|} \frac{(\widetilde{N}_m)! \left(N_m^i + |\Sigma| - 1\right)!}{\left(\widetilde{N}_m + N_m^i + |\Sigma| - 1\right)!} \\ &\times \prod_{n=1}^{|\Sigma|} \frac{(\widetilde{N}_{mn} + N_{mn}^i)!}{(\widetilde{N}_{mn})!(N_{mn}^i)!} \end{aligned} \tag{12}$$

In practice, it might be easier to compute the logarithm of $\Pr(\widetilde{S}|S^i)$ by using Stirling's approximation formula $\log(n!) \approx n\log(n) - n$ [28] because, in most cases, both N^i and $\widetilde{N}$ would consist of statistically large

enough numbers (but still not be enough to directly estimate a Π at the testing phase).

B. Class Assignment of a Slow Time Epoch

After the inference step, the probability that $\widetilde{S}$ is assigned to an existing class C^i (in the set of existing classes $\mathbf{C} = \{C^1, C^2, \ldots, C^K\}$) need to be determined. Essentially, the quasi-stationary characteristics demonstrated by S^i is denoted by C_i. From inference computation, the likelihood function $\Pr(\widetilde{S}|S^i)$ is obtained and can also be written as $Pr(\widetilde{S}|C^i)$. The posterior $Pr(C^i|\widetilde{S})$ then can be expressed as

$$Pr(C^i|\widetilde{S}) \propto Pr(\widetilde{S}|C^i) \times Pr(C^i) \tag{13}$$

where $\Pr(C^i)$ denotes a known prior for class C_i. Now, in the current unsupervised context knowing prior for a class may not be possible. However, when a time-series segment at slow time epoch $\tau - 1$ belongs to class C^j, then the probability that data at epoch τ will belong to the same class C^j may be higher compared to probabilities for other classes. This is a realistic assumption as a real system typically may not change operating point or parametric condition for every slow time epoch. As a consequence the distribution $Pr(C^i)$ at τ can be considered skewed in favor of C^j. With K existing classes this is realized as

$$Pr(C^i) = \begin{cases} \frac{\kappa}{K-1+\kappa} & \text{for } i = j \\ \frac{1}{K-1+\kappa} & \text{for } i \neq j \end{cases}$$

where $\kappa > 1$ as a stickiness factor.

After computing $Pr(\widetilde{S}|C^i)Pr(C^i)$ for all K existing classes, a decision is made regarding the class assignment using CRP as mentioned in Section 2-B. Naturally, the concentration or strength function $\mathbb{C}(\cdot)$ is chosen as

$$\mathbb{C}(C^i) = Pr(\widetilde{S}|C^i)Pr(C^i) \tag{14}$$

Therefore, the CRP formulation with parameter γ can be written as

$$Pr_\gamma(C^i|\widetilde{S}) = \begin{cases} \frac{\mathbb{C}(C^i)}{[\sum_{C_j \in \mathbf{C}} \mathbb{C}(C_j)]+\gamma} & \text{for } i = 1, \cdots, K \\ \frac{\gamma}{[\sum_{C_j \in \mathbf{C}} \mathbb{C}(C_j)]+\gamma} & \text{for } i = K+1 \end{cases} \tag{15}$$

where, $Pr_\gamma(C^i|\widetilde{S})$ is the probability with which $\widetilde{S}$ is assigned to class C^i and C^{K+1} is a new unforeseen class. The online algorithm for learning Tier 1 PFSAs is summarized below:

Algorithm 1: Online Learning of Tier 1 PFSAs

Input Parameters: Stickiness parameter κ and CRP parameter γ
Data Input: Symbol sequence segments $\widetilde{S}_{\tau_l}$ for slow time epochs $\tau_1, \tau_2, \cdots$
Initialize: $\mathbf{C} = \{C^1\}$
Initialize: All $N^1_{mn} = 0$ (m, n chosen based on $|Q|$ and $|\Sigma|$)
Compute N^1_{mn} using $\widetilde{S}_{\tau_1}$
FORALL $\tau_2, \tau_3, \cdots$ **DO**
  Compute $\widetilde{N}_{mn}$ using $\widetilde{S}_{\tau_l}$
  Evaluate $Pr_\gamma(C^i|\widetilde{S}_{\tau_l})$ using Eqn. 3-B $\forall C^i \in \mathbf{C} = \{C^1, C^2, \ldots, C^K\}$ and C^{K+1}
  Assign $\widetilde{S}_{\tau_l}$ to a class C^j according to Pr_γ
  IF $j \in \{1, 2, \ldots, K\}$
   Update N^j_{mn} by appending $\widetilde{S}_{\tau_l}$ to S^j
  ELSEIF $j = K + 1$
   Update $\mathbf{C}$ as $\{C^1, C^2, \ldots, C^K, C^{K+1}\}$
   Compute N^{K+1}_{mn} using $\widetilde{S}_{\tau_l}$
  ENDIF
ENDFOR

The algorithm described above identifies different classes of quasi-stationary characteristics in an online fashion and Tier 1 PFSAs can represent those characteristics by different Π matrices. However, due to noise and spurious disturbances present in data, redundant classes may appear during the online learning process. Therefore, a periodic (with a much slower time-scale, i.e., after many slow time epochs), a revision step can be included to merge different PFSAs that are close enough based on a metric [29] defined below.

Definition 3.1: (Distance Metric for PFSA) Let $\mathcal{P}_1 = (Q_1, \Sigma, \delta_1, \Pi_1$ and $\mathcal{P}_2 = (Q_2, \Sigma, \delta_2, \Pi_2$ be two PFSA with a common alphabet Σ. Let $P_1(\Sigma^j)$ and $P_2(\Sigma^j)$ be the steady state probability vectors of generating words of length j from the PFSA $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, i.e., $P_1(\Sigma^j) \triangleq \triangle[P(w)]_{w \in \Sigma^j}$ for $\mathcal{P}_1$ and $\mathcal{P}_2 \triangleq [P(w)]_{w \in \Sigma^j}$ for $\mathcal{P}_2$. Then, the metric for the distance between the PFSAs $\mathcal{P}_1$ and $\mathcal{P}_2$ is defined as

$$\Phi(\mathcal{P}_1, \mathcal{P}_2) \triangleq \lim_{n \to \inf} \sum_{j=1}^{n} \frac{\|P_1(\Sigma^j) - P_2(\Sigma^j)\|_{l_1}}{2^{j+1}} \tag{16}$$

where the norm $\| \star \|_{l_1}$ indicates the sum of absolute values of the elements in the vector $\star$.
Thus, the revision step can merge two Tier 1 PFSAs $\mathcal{P}_1$ and $\mathcal{P}_2$ when $\Phi(\mathcal{P}_1, \mathcal{P}_2) < \eta$, where $\eta > 0$ is a suitable threshold for checking similarity. In this paper, only symbols (i.e., words of length 1) have been considered for calculating the above metric. As Tier 1 PFSAs get identified online, the Tier 2 PFSA can be learnt simply by keeping track of the transitions of the system from one Tier 1 PFSA to another.

4. VALIDATION

A. Simulated Duffing System

The Duffing equation is a nonlinear active system that shows chaotic behavior [30]. It can be implemented on a real experimental test bed involving an electronic circuit [19]. The equation is stated as:

$$\frac{d^2x(t)}{dt^2} + \beta\frac{dx(t)}{dt} + \alpha_1 x(t) + x^3(t) = Acos(\omega t) \quad (17)$$

where A = 22.0 is the amplitude of the forcing function, ω = 5.0 rad/s is its excitation frequency and α_1 = 1.0 and the dissipation parameter β. Variation of β is known to change the system characteristics and a sudden mode shift happens around $\beta = 0.3$ [11]. Therefore, this system is considered with two β values 0.1 and 0.4 (one before bifurcation and one after) to generate two different quasi-stationary classes. With these two classes a streaming non-stationary time-series is generated by randomly selecting $\beta \in \{0.1, 0.4\}$ for different segments. A phase plot of the output y vs the forcing function for a typical non-stationary data set is shown in Fig. 2.

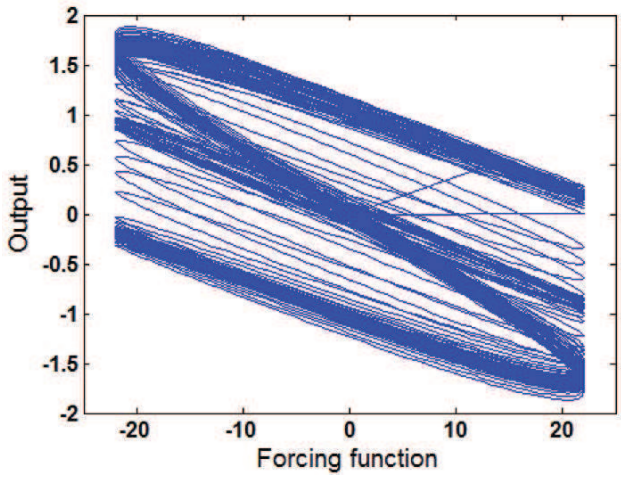

Fig. 2. Phase plot of non-stationary dynamics for Duffing System

B. Results and Discussion

A randomly generated (as described above) time-series of y with 400000 has been used for the results provided here. As every slow time epoch is considered to have 1000 points, the accuracy plots are based on 400 epochs. The raw time-series is symbolized using 8 uniform data space partitions. Figure 3 demonstrates the performance of Hierarchical SDF (HSDF) algorithm by plotting class labels from ground truth and HSDF for streaming data epochs. It is observed that a lot of redundant classes are created due to noise and spurious disturbances present in the data when the decision system uses only CRP. The performance improves significantly when stickiness factor is imposed, i.e., number of uniquely identified classes come down from 22 to 5. However, with close observation it can be seen that class 2 actually gets split between class $2, 3, 4$ and 5 during online learning process. And as a matter of fact they are represented by PFSAs that are very similar. Therefore, the periodic revision process can easily merge them to obtain nearly perfect accuracy by identifying not more than 2 classes.

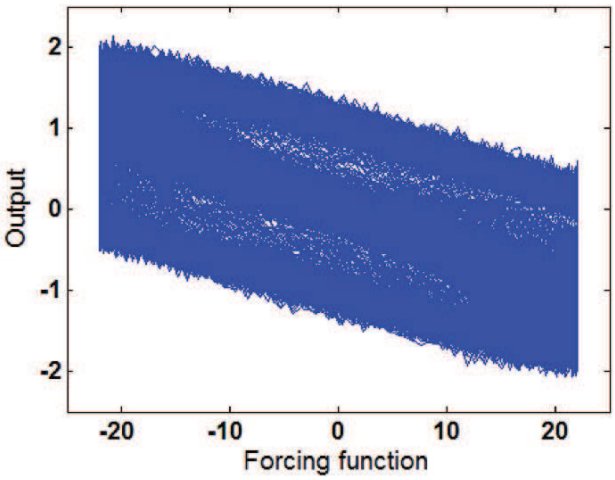

Fig. 4. Phase plot of non-stationary dynamics for Duffing System with SNR =10

One of the advantages of SDF-based tools is that primarily due to the partitioning process, it is typically very robust to the change in noise characteristics in the data. To investigate similar property of the present hierarchical extension algorithm the experiments are repeated with significant increase in noise content of the data. The same time-series is considered now with signal to noise ratio (SNR) as 10. The phase plot in Fig. 4 shows the significant increase in noise compared to that in Fig. 2. Figure 5 demonstrates the performance of HSDF before and after revision. The algorithm obtains similar accuracy after revision and only a few more (increased from 5 to 8 classes) redundant classes appear before revision. For all the results shown here, a few manual iterations were required to choose the correct set of hyper-parameters (i.e., CRP parameter γ, stickiness parameter κ and revision threshold η) to achieve the demonstrated performance. Therefore, automated selection of hyper-parameters would be the next technical problem for investigation. Also, quantitative performance metrics will be defined in order to obtain numerical comparison purposes. Also, the algorithm could achieve this performance in real-time with a simple MATLAB implementation on a 3.40 GHZ Intel Xeon(R) CPU with Windows OS and 16GB RAM.

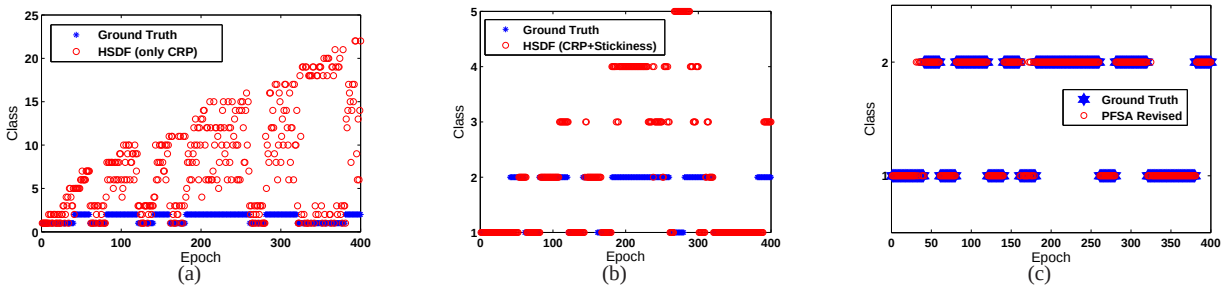

Fig. 3. Plots of class labels from ground truth and Hierarchical SDF (HSDF) for streaming data epochs; Plate (a) shows performance using only CRP, Plate (b) shows performance improvement with use of stickiness factor and Plate (c) shows the best performance with periodic revision

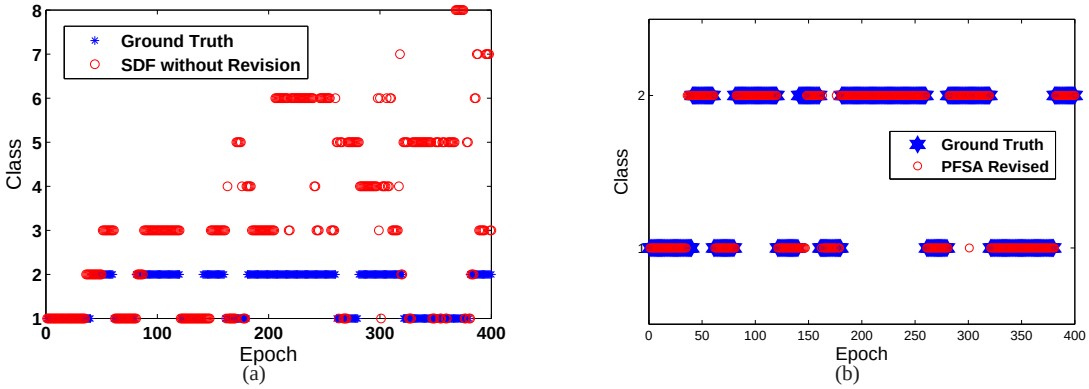

Fig. 5. Plots of class labels from ground truth and Hierarchical SDF (HSDF) for streaming data epochs with increased noise content, SNR = 10; Plate (a) shows performance before the revision step and Plate (b) shows the performance with periodic revision

5. Summary, Conclusions and Future work

This paper extends the concepts of Symbolic Dynamic Filtering (SDF) of quasi-stationary time-series to develop a hierarchical feature extraction technique for non-stationary time-series data. In the present context, a non-stationary time-series is considered to be a slow-time scale mixture of time-series segments that are quasi-stationary at a faster time-scale. While PFSAs at the lower level capture the fast time-scale quasi-stationary dynamics, a PFSA at the upper level capture the slow time-scale transitions of the system among different quasi-stationary dynamics. The algorithm developed here is an unsupervised tool that allows to analyze data with an unknown number of unique quasi-stationary characteristics (or the number of states for the upper level PFSA). Essentially, the learning process uses a Bayesian inference scheme for short data length classification using SDF for a new segment of streaming data. The inference process determines whether the data segment belongs to an already existing quasi-stationary class or it represents an unforeseen characteristics. The assignment decision-making process also involves a Chinese Restaurant Process (CRP) along with a stickiness factor. The online algorithm is validated using a nonlinear active electronic system based on the chaotic Duffing equation. It has been shown that a simple revision algorithm can be used periodically to merge different PFSAs at the lower level that are very similar to each other to reduce unnecessary model complexity. While further tests using real-life data sets are being performed to evaluate the efficacy of the algorithm, the major future theoretical research directions are mentioned below.

- *Formulation of hierarchical PFSA learning as an optimization problem with model accuracy and complexity as competing objectives; this will enable optimization of hyper-parameters (such as, CRP parameter γ and stickiness parameter κ) and provide quantitative justification for hierarchical models instead of single tier ones*
- *Currently, homogeneous PFSAs (i.e., with same structure) are learnt at the lower level. Future research will investigate adaptive PFSA learning at the lower level to fit the exact need of different quasi-stationary characteristics*
- *Performance comparison (i.e., accuracy, computation time and complexity) with other hierarchical feature extraction tools using benchmark data sets*
- *Extension of the algorithm to fuse multiple time-series information during feature extraction*

REFERENCES

[1] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," in *IEEE International Conference on Robotics and Automation (ICRA)*, May, 2014.

[2] C. S. Duarte Antunes, J. Hespanha, "Stochastic hybrid systems with renewal transitions: Moment analysis with applications to networked control systems with delays," *SIAM J. Contr. Optimization*, vol. 51, no. 2, pp. 1481–1499, 2013.

[3] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313.5786, pp. 504–507, 2006.

[4] Y. Bengio and D. Olivier, "On the expressive power of deep architectures," *Algorithmic Learning Theory. Springer Berlin/Heidelberg*, 2011.

[5] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep boltzmann machines," *Neural Computation*, vol. 24, no. 8, pp. 1967–2006, 2012.

[6] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "A sticky hdp-hmm with application to speaker diarization," *The Annals of Applied Statistics*, vol. 5, no. 2A, pp. 1020–1056, 2011.

[7] S. Tranter and D. Reynolds, "An overview of automatic speaker diarisation systems," *IEEE Transactions on Speech, Audio and Language Processing: Special Issue on Rich Transcription*, pp. 1557–1565, 2006.

[8] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, "How to grow a mind: Statistics, structure, and abstraction," *Science*, vol. 331, pp. 1279–1285, 2011.

[9] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.

[10] S. Sarkar, S. Sarkar, K. Mukherjee, A. Ray, and A. Srivastav, "Multi-sensor data interpretation and semantic fusion for fault detection in aircraft gas turbine engines," *Proceedings of the I Mech E Part G: Journal of Aerospace Engineering*, vol. 227, no. 12, pp. 1988–2001, December 2013.

[11] C. Rao, A. Ray, S. Sarkar, and M. Yasar, "Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns," *Signal, Image and Video Processing*, vol. 3, no. 2, pp. 101–114, 2009.

[12] X. Jin, Y. Guo, S. Sarkar, A. Ray, and R. Edwards, "Anomaly detection in nuclear power plants via symbolic dynamic filtering," *IEEE Transactions on Nuclear Science*, vol. 58, no. 1, pp. 277–288, February 2011.

[13] S. Chakraborty, S. Sarkar, S. Gupta, and A. Ray, "Damage monitoring of refractory wall in a generic entrained-bed slagging gasification system," *Proceedings of the I Mech E Part A: Journal of Power and Energy*, vol. 222, Part A, no. 8, pp. 791–807, October 2008.

[14] S. Sarkar, N. Virani, M. Yasar, A. Ray, and S. Sarkar, "Proceedings of american control conference, washington, dc," in *Spatiotemporal Information Fusion for Fault detection in Shipboard Auxilliary Systems*, 2013.

[15] S. Sarkar, M. Yasar, S. Gupta, A. Ray, and K. Mukherjee, "Fault detection and isolation in aircraft gas turbine engines: Part ii - validation on a simulation test bed," *Proceedings of the I Mech E Part G: Journal of Aerospace Engineering*, vol. 222, no. 3, pp. 319–330, May 2008.

[16] S. Gupta, A. Ray, S. Sarkar, and M. Yasar, "Fault detection and isolation in aircraft gas turbine engines: Part i - underlying concept," *Proceedings of the I Mech E Part G: Journal of Aerospace Engineering*, vol. 222, no. 3, pp. 307–318, May 2008.

[17] S. Sarkar, S. Sarkar, and A. Ray, *Data-enabled Health Management of Complex Industrial Systems*. Fault Detection: Classification, Techniques and Role in Industrial Systems, NOVA Science Publishers, 2014.

[18] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Sharing features among dynamical systems with beta processes," PROCEEDINGS OF NEURAL INFORMATION PROCESSING SYSTEMS, 2009.

[19] C. Rao, K. Mukherjee, S. Sarkar, and A. Ray, "Statistical estimation of multiple parameters via symbolic dynamic filtering," *Signal Processing*, vol. 89, pp. 981–988, June 2009.

[20] S. Sarkar, A. Srivastav, and M. Shashanka, "Maximally bijective discretization for data-driven modeling of complex systems," in *Proceedings of American Control Conference, Washington, D.C.*, 2013.

[21] S. Sarkar, K. Mukherjee, X. Jin, and A. Ray, "Optimization of symbolic feature extraction from time-series for classification," *Signal Processing*, vol. 92, no. 3, pp. 625–635, March 2012.

[22] Y. Whye Teh, "A tutorial on dirichlet processes and hierarchical dirichlet processes," March 2007.

[23] D. J. Aldous, *Exchangeability and related topics in Lecture Notes in Mathematics*, vol. 1117. Springer Berlin Heidelberg, 1985.

[24] S. Sarkar, K. Mukherjee, S. Sarkar, and A. Ray, "Symbolic dynamic analysis of transient time series for fault detection in gas turbine engines," Technical Brief DS-11-1309, The Pennsylvania State University, University Park, PA 16802, 2000. To appear in J. Dyn. Sys. Meas. Control.

[25] T. Ferguson, "Ea bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, 1973.

[26] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, 1994.

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

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

[29] K. Mukherjee and A. Ray, "State splitting and merging in probabilistic finite state automata for signal representation and analysis," *Signal processing*, vol. 104, pp. 105 –119, 2014.

[30] R. V., S. Chakraborty, and A. Ray, "Estimation of slowly varying parameters in nonlinear systems via symbolic dynamic filtering," *Signal Processing*, no. 88, pp. 339 – 348, 2008.