Signal Processing 104 (2014) 105–119

Contents lists available at ScienceDirect

Signal Processing

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

State splitting and merging in probabilistic finite state automata for signal representation and analysis ☆

Kushal Mukherjee [a,1], Asok Ray [b,*]

[a] United Technology Research Center, Cork, Ireland
[b] The Pennsylvania State University, University Park, PA, USA

ARTICLE INFO

Article history:
Received 27 July 2013
Received in revised form
24 March 2014
Accepted 28 March 2014
Available online 4 April 2014

Keywords:
Probabilistic finite state automata
Symbolic dynamics
D-Markov machines
State splitting
State merging

ABSTRACT

Probabilistic finite state automata (PFSA) are often constructed from symbol strings that, in turn, are generated by partitioning time series of sensor signals. This paper focuses on a special class of PFSA, which captures *finite history* of the symbol strings; these PFSA, called *D*-Markov machines, have a simple algebraic structure and are computationally efficient to construct and implement. The procedure of PFSA construction is based on (i) *state splitting* that generates symbol blocks of different lengths based on their information contents; and (ii) *state merging* that assimilates histories by combining two or more symbol blocks without any significant loss of the embedded information. A metric on the probability distribution of symbol blocks is introduced for trade-off between loss of information (e.g., entropy rate) and the number of PFSA states. The underlying algorithms have been validated with three test examples. While the first and second examples elucidate the key concepts and the pertinent numerical steps, the third example presents the results of analysis of time series data, generated from laboratory experimentation, for detection of fatigue crack damage in a polycrystalline alloy.

1. Introduction

Symbolic time series analysis (STSA) [1,2] is built upon the concept of symbolic dynamics [3] that deals with discretization of dynamical systems in both space and time. The notion of STSA has led to the development of a pattern recognition tool, in which a time series of sensor signals is represented as a symbol sequence that, in turn, leads to the construction of probabilistic finite state automata (PFSA) [4–9]. The paradigm of PFSA has been used for behavior modeling of dynamical systems and its applications are widespread in various fields including computational linguistics [10] and speech recognition [11]. Since PFSA models are capable of efficiently compressing the information embedded in sensor time series [12,13], these models could enhance the performance and execution speed of information fusion [14] and information source localization [15] that are often computation-intensive. Rao et al. [16], Jin et al. [17] and Bahrampour et al. [18] have shown that the performance of this PFSA-based tool as a feature extractor for statistical pattern recognition is comparable (and often superior) to that of other existing techniques (e.g., Bayesian filters, Artificial Neural Networks, and Principal Component Analysis [19]).

Statistical patterns of slowly evolving dynamical behavior in physical processes can be identified from sensor time series data [1]. Often the changes in these statistical

☆ This work has been supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant nos. W911NF-07-1-0376 and W911NF-13-1-0461, and by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant no. FA9550-12-1-0270. 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: MukherK@utrc.utc.com (K. Mukherjee), axr2@psu.edu (A. Ray).
[1] Formerly with The Pennsylvania State University, University Park, PA, USA.

http://dx.doi.org/10.1016/j.sigpro.2014.03.045
0165-1684/

patterns occur over a slow time scale with respect to the fast time scale of process dynamics. In this context, the concept of two time scales is succinctly presented below.

Definition 1.1 (*Fast scale*). The fast scale is defined to be a time scale over which the statistical properties of the process dynamics are assumed to remain invariant, i.e., the process is assumed to have statistically stationary dynamics at the fast scale.

Definition 1.2 (*Slow scale*). The slow scale is defined to be a time scale over which the statistical properties of the process dynamics may gradually evolve, i.e., the process may exhibit statistically non-stationary dynamics at the slow scale.

In view of Definition 1.1, statistical variations in the internal dynamics of the process are assumed to be negligible at the fast scale. Thus, sensor time series data are acquired based on the assumption of statistical stationarity at the fast scale. In view of Definition 1.2, an observable non-stationary behavior could be associated with the gradual evolution of anomalies (i.e., deviations from the nominal behavior) in the process at the slow scale. In general, a long time span at the fast scale is a tiny (i.e., several orders of magnitude smaller) interval at the slow scale. A pictorial view of the two-time-scales operation in Fig. 1 illustrates the concept.

The major steps for construction of PFSA from sensor signal outputs (e.g., time series) of a dynamical system are

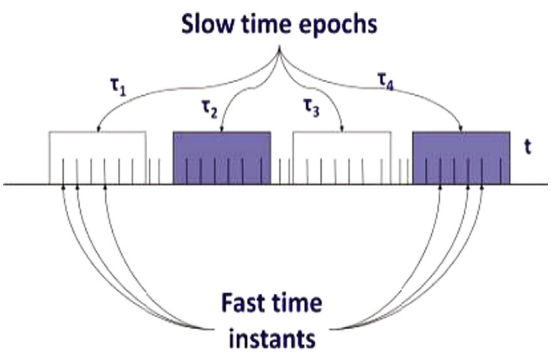

Fig. 1. Underlying concept of fast and slow time scales.

as follows:

(1) Coarse-graining of time series to convert the scalar or vector-valued data into symbol strings, where the symbols are drawn from a (finite) alphabet [20].
(2) Encoding of probabilistic state machines from the symbol strings [12,21].

In the process of symbol generation, the space of time series is partitioned into finitely many mutually exclusive and exhaustive cells, each corresponding to a symbol belonging to a (finite) alphabet. As a trajectory of the dynamical system passes through or touches various cells of the partition, the symbol assigned to the cell is inserted in the symbol string. In this way, a time series corresponding to a trajectory is converted into a symbol string. Fig. 2 illustrates the concept of constructing finite state automata (FSA) from time series, which provides the algebraic structure of probabilistic finite state automata (PFSA).

The next step is to construct probabilistic finite state automata (PFSA) from the symbol strings to encode their statistical characteristics so that the dynamical system's behavior is captured by the patterns generated by the PFSA in a compact form. The algebraic structure of PFSA (i.e., the underlying FSA) consists of a finite set of states that are interconnected by transitions [22–24], where each transition corresponds to a symbol in the (finite) alphabet. At each step, the automaton moves from one state to another (possibly including self loops) via these transitions, and thus generates a corresponding block of symbols so that the probability distributions over the set of all possible strings defined over the alphabet are represented in the space of PFSA. The advantage of such a representation is that the PFSA structure is simple enough to be encoded as it is characterized by the set of states, the transitions (i.e., exactly one transition for each symbol generated at a state), and the transition's probability of occurrence.

D-Markov machines are models of probabilistic languages where the future symbol is causally dependent on the (most recently generated) finite set of (at most) *D* symbols and form a proper subclass of PFSA with applications in various fields of research such as anomaly detection [12] and robot motion classification [25]. The

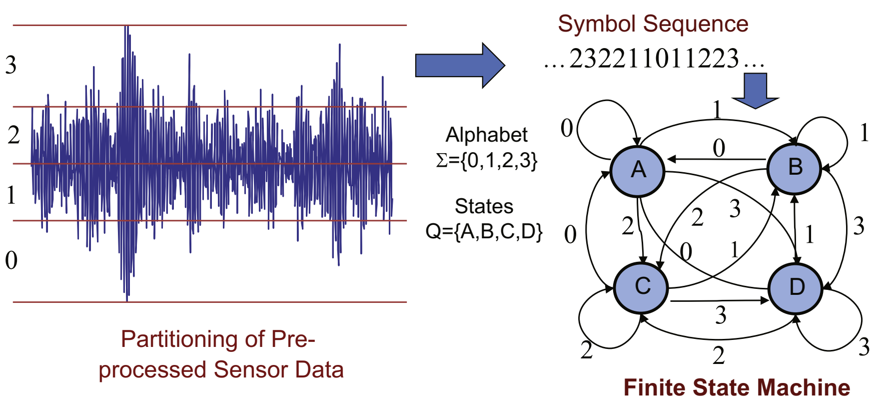

Fig. 2. Construction of probabilistic finite state automata (PFSA).