

Dynamic Data-driven Symbolic Causal Modeling for Battery Performance & Health Monitoring

Soumalya Sarkar[†] Devesh K. Jha[†] Asok Ray Yue Li
svs5464@psu.edu dkj5042@psu.edu axr2@psu.edu yol5214@psu.edu

Pennsylvania State University, University Park, PA 16802, USA

[†] Authors with equal contribution

Abstract—The paper presents a dynamic data-driven symbolic approach to construct generative models of causal cross-dependence among different sources of (possibly heterogeneous) measurements. The main objective here is to identify the input-output relationships in the underlying dynamical system using sensory data only. Synchronized pairs of input and output time series are first independently symbolized via partitioning the individual data sets in their respective range spaces. A generative model is then obtained to capture cross-dependency in the symbolic input-output dynamics as a variable-memory cross D-Markov (also called xD -Markov) machine, which is different from the standard PFSA. The proposed input-output model has been validated on charging-discharging data sets of a lead-acid battery. The cross-dependency features of current-voltage patterns during charging-discharging cycles have been used to estimate and predict the parameters of battery performance (e.g., State-of-Charge (SOC)) and health (e.g., State-of-Health (SOH)).

Keywords: Sensor Fusion; Input-output Characterization; Symbolic Time Series Analysis; Battery Health Monitoring

1. MOTIVATION AND INTRODUCTION

Modeling causal dependence between variables, events or physical processes is important in almost all data-driven scientific inquiry to discern statistical relationships between them. It finds applications in various fields like statistics, physics, medicine, economics and more recently, machine learning [1], [2], [3], [4]. Much of the work available in literature has been focused on defining measures to test existence of a causal relationship between two stochastic processes [5], [6], [7]. In this regard, information-theoretic measures have been defined to establish and test existence of causality among observed variables (e.g., transfer entropy and directed information) [3], [4], [8], [9], [10]. However, most of the work is focused on measuring the degree of causal dependence between stochastic processes. Apparently, very little work has been presented to infer the causal generative structure between two observed variables that can be used for prediction and estimation of a dependent variable, based on the observations of the other. We present a symbolic analysis

based statistical modeling approach to construct generative models of causal cross-dependence between two stochastic processes.

Symbolic time-series analysis (STSA) is a non-linear statistical tool for modeling temporal patterns in sequential data. Symbolic analysis of time-series for precise modeling of the underlying dynamics must simultaneously satisfy the following two criteria.

- 1) *Symbolization*: The process of projecting continuously-varying time-series data onto a symbol space.
- 2) *Depth Estimation*: The task of estimating the temporal memory of the underlying dynamical system.

Once the data sets are symbolized and the depth (i.e., memory) for the symbol sequence is estimated, the symbol stream can be compressed into generative models as probabilistic finite state automata (PFSA). A class of PFSAs, called the D-Markov machine, has been proposed as a sub-optimal but computationally efficient approach to encode the statistical behavior of a symbol stream having the algebraic structure of a deterministic finite state machine (DFSA) [11], [12].

This paper presents a method of inferring generative models of causal dependence between two observed variables, where the generative model is built upon a Markov structure between the observed variables. A causal cross-dependence between the two synchronized data streams is represented as *crossed automata* known as cross D-Markov (xD -Markov) machines which was introduced earlier in [13]. The motivation is to be able to predict the behavior of a dynamical system using observations on the input states. The Markov dynamics apply to the input states for prediction on output states as opposed to the output states themselves. In an input-output setting, this approach becomes analogous to transfer function representation for finite-dimensional linear time-invariant (FDLTI) systems studied in the classical control theory, which has been routinely used in different fields (e.g., health-monitoring, cyber-security, and power grids) for online system identification and fault/anomaly detection.

In the D-Markov setting, the equivalence class of states

(i.e., strings of symbols) is inferred for a symbol stream to make predictions on the probabilities of symbol emission for the the other stream. A key difference from most of the work, reported in technical literature, is that this paper only considers the dependence of a symbol sequence on the other (synchronized) symbol sequence, instead of joint observations on both of them. This approach greatly alleviates the complexity as there is no need to create the product state-space for the two symbol sequences.

The proposed approach is similar to mixed-memory Hidden Markov Models (HMMs) which models a complex stochastic process as a mixture of simpler processes [14]. However, instead of considering a weighted combination of the cross-transition matrices [14], [13] to accurately model the Markov dynamics, this paper makes use of directed information contents between the observed variables to construct a variable-memory cross-transition model. This approach has the potential benefit of reducing the model complexity without any significant compromise of modeling details; the rationale is that there is no need to separately infer the models with different memories.

Contributions: To the best of the authors' knowledge, this paper, for the first time, introduces a variable-memory xD -Markov machine to model the causal cross-dependency between two time-series. Under the assumptions of statistical stationarity of the underlying data, the variable-memory structure in the xD -Markov machine setting [13], [15] is inferred using entropy rate as the metric to measure the directed information content of the cross-dependency model. Using only the sequential observations on the two variables, we infer the non-heuristic generative models of cross-dependence without any assumptions on the nature of the hidden dynamics, linear or non-linear. Furthermore, the proposed algorithm has been validated using experimental data for a lead-acid battery which is charged and discharged using variable input current. The xD -Markov setting is used to learn the input-output relationship for a lead-acid battery. These relationships are then used to estimate the parameters of battery performance (e.g., State-of-Charge (SOC)) and health (e.g., State-of-Health (SOH)).

Organization: The paper is organized in seven sections including the current one. Section 2 briefly outlines the pertinent mathematical concepts that serve as foundations of the work presented in this paper. Section 3 presents the proposed approach to infer the hidden cross-dependence between two observed time series in the PFSA setting. Section 4 describes the procedure for acquisition of time series data from a lead-acid battery. Section 5 presents xD -Markov modeling of the input-output characteristics of the battery dynamics. Section 6 presents the results on identification of SOH and SOC parameters of the battery system. Finally the paper is summarized and concluded in

Section 7.

2. MATHEMATICAL PRELIMINARIES

This section introduces mathematical concepts that serve as foundations of the work presented in this paper.

Definition 2.1 (PFSA [11], [12]) *A probabilistic finite state automaton (PFSA) is constructed on the algebraic structure of deterministic finite state automata (DFSA) $G = (\Sigma, Q, \delta)$ as a pair $K = (G, \pi)$, i.e., the PFSA K is a 4-tuple $K = (\Sigma, Q, \delta, \pi)$, where:*

- 1) Σ is a non-empty finite set, called the symbol alphabet, with cardinality $|\Sigma| < \infty$;
- 2) Q is a non-empty finite set, called the set of states, with cardinality $|Q| < \infty$;
- 3) $\delta : Q \times \Sigma \rightarrow Q$ is the state transition map;
- 4) $\tilde{\pi} : Q \times \Sigma \rightarrow [0, 1]$ is the symbol generation function (also called probability morph function) that satisfies the condition $\sum_{\sigma \in \Sigma} \tilde{\pi}(q, \sigma) = 1 \quad \forall q \in Q$, and π_{ij} is the probability of emission of a symbol $\sigma_j \in \Sigma$ when the state $q_i \in Q$ is observed.

Definition 2.2 (D -Markov [11], [12]) *A D -Markov machine is a statistically stationary stochastic process $S = \cdots s_{-1}s_0s_1\cdots$ (modeled by a PFSA in which each state is represented by a finite history of D symbols), where the probability of occurrence of a new symbol depends only on the last D symbols, i.e.,*

$$P[s_n | \cdots s_{n-D} \cdots s_{n-1}] = P[s_n | s_{n-D} \cdots s_{n-1}] \quad (1)$$

- D is called the depth of the Markov machine;
- Q is the finite set of states with cardinality $|Q| \leq |\Sigma|^D$, i.e., the states are represented by equivalence classes of symbol strings of maximum length D .
- $\delta : Q \times \Sigma \rightarrow Q$ is the state transition function.

Definition 2.3 (Symbol Block [12]) *A symbol block, also called a word, is a finite-length string of symbols belonging to the alphabet Σ , where the length of a word $w \triangleq s_1s_2\cdots s_\ell$ with $s_i \in \Sigma$ is $|w| = \ell$, and the length of the empty word ϵ is $|\epsilon| = 0$.*

- The set of all words constructed from symbols in Σ , including the empty word ϵ , is denoted as Σ^* ,
- The set of all words, whose suffix (respectively, prefix) is the word w , is denoted as Σ^*w (respectively, $w\Sigma^*$).
- The set of all words of (finite) length ℓ , where $\ell > 0$, is denoted as Σ^ℓ .

Definition 2.4 (Conditional Entropy [12]) *The entropy of a PFSA (Σ, Q, δ, π) conditioned on the current state $q \in Q$ is defined as follows.*

$$H(\Sigma|q) \triangleq - \sum_{\sigma \in \Sigma} P(\sigma|q) \log P(\sigma|q) \quad (2)$$

where $P(\sigma|q)$ is the conditional probability of a symbol $\sigma \in \Sigma$ given that a PFSA state $q \in Q$ is observed.

Definition 2.5 (Entropy Rate [12]) *The entropy rate of a PFSA (Σ, Q, δ, π) is defined in terms of the conditional entropy as follows.*

$$\begin{aligned} H(\Sigma|Q) &\triangleq \sum_{q \in Q} P(q)H(\Sigma|q) \\ &= - \sum_{q \in Q} \sum_{\sigma \in \Sigma} P(q)P(\sigma|q) \log P(\sigma|q) \quad (3) \end{aligned}$$

where $P(q)$ is the probability of a PFSA state $q \in Q$.

The entropy rate represents the overall predictability of a PFSA.

3. PROPOSED APPROACH

This section presents the proposed approach to infer the hidden cross-dependence between two observed time series in the PFSA setting. Previous attempts on modeling cross-dependence between symbol sequences using xD -Markov machines have focused on building a fixed depth model, where the depth parameter is either manually selected or searched using a wrapper-like method while cross-dependence between the symbolic processes is measured using mutual information [15]. Apparently, such information measures have never been used to infer the generative structure between the sequences. Formally, the xD -Markov machine is defined as follows:

Definition 3.1 (xD -Markov) *Let \mathcal{M}_1 and \mathcal{M}_2 be the PFSA's corresponding to symbol streams $\{s_1\}$ and $\{s_2\}$, respectively. Then, an xD -Markov machine (from $\{s_1\}$ to $\{s_2\}$) is defined as a 5-tuple $\mathcal{M}_{1 \rightarrow 2} \triangleq (\mathcal{Q}_1, \Sigma_1, \Sigma_2, \delta_1, \Pi_{12})$ such that:*

- $\mathcal{Q}_1 = \{q_1, q_2, \dots, q_{|\mathcal{Q}_1|}\}$ is the state set corresponding to symbol sequence $\{s_1\}$
- $\Sigma_1 = \{\sigma_0^1, \dots, \sigma_{|\Sigma_1|-1}^1\}$ is the alphabet set of symbol sequence $\{s_1\}$
- $\Sigma_2 = \{\sigma_0^2, \dots, \sigma_{|\Sigma_2|-1}^2\}$ is the alphabet set of symbol sequence $\{s_2\}$
- $\delta_1 : \mathcal{Q}_1 \times \Sigma_1 \rightarrow \mathcal{Q}_1$ is the state transition mapping for \mathcal{M}_1
- Π_{12} is the cross morph matrix of size $|\mathcal{Q}_1| \times |\Sigma_2|$; the ij^{th} element $(\pi_{12}(q_i, \sigma_j^2))$ of Π_{12} denotes the probability of finding the symbol σ_j^2 in the symbol string $\{s_2\}$ at next time step while making a transition from the state q_i of the PFSA constructed from the symbol sequence $\{s_1\}$.

The states of the PFSA's, \mathcal{M}_1 and \mathcal{M}_2 , in Definition 3.1 are constructed by symbols when those individual PFSA's are represented by D-Markov machines. The D-Markov

machines with depth $D = 1$ represent the states as symbols of the corresponding alphabet set. For $D > 1$, the states of the D-markov machines are obtained via state splitting and merging method [12] to restrict the dimensionality of state space while modeling the essential dynamics of the PFSA. State splitting is based on the process of minimizing the entropy rate of PFSA. However, the states of PFSA \mathcal{M}_1 may not be able to capture the cross-dependence from symbol strings $\{s_1\}$ to $\{s_2\}$ optimally via an xD -Markov machine as they are modeled based only on $\{s_1\}$. This paper proposes a method to construct the state space of \mathcal{M}_1 such that it captures the optimal cross-dependence via Π_{12} and yields better input-output characterization of dynamical systems (e.g., a battery system).

A. Time series symbolization

The first step in the construction of an optimal xD -Markov machine from one time-series to another is to symbolize them. This step requires independent partitioning of the two time series. The signal space of a time-series is partitioned into a finite number of cells that are labeled as symbols, i.e., the number of cells is identically equal to the cardinality of the (symbol) alphabet. These mutually exclusive and exhaustive regions form a partition, where each region is labeled with one symbol from the alphabet sets Σ_1 (for first time series) or Σ_2 (for second time series). If the value of time series at a given instant is located in a particular cell, then it is coded with the symbol associated with that cell [16]. Thus, finite arrays of symbols $\{s_1\}$ and $\{s_2\}$, called symbol strings, are generated from the two (finite-length) time series data.

There are different types of partitioning tools such as maximum entropy partitioning (MEP), uniform partitioning (UP) [16] and supervised optimal partitioning [17]. MEP maximizes the entropy of the generated symbols by putting (approximately) equal number of data points in each partition cell and therefore, the information-rich portions of a time-series are partitioned finer and those with sparse information are partitioned coarser. In UP, the partitioning lines create uniform (by size on signal space) cells.

B. Algorithm Development

This subsection develops the algorithm for construction of xD -Markov machines. For further analysis, a type of entropy rate (Eq. (3)) called cross entropy rate is defined here to represent the overall predictability of a symbol stream when a PFSA from another symbol stream is observed. The cross entropy rate from a PFSA

$(\Sigma_1, Q_1, \delta_1, \pi_1)$ to a symbol stream (say, $\{s_2\}$ with alphabet set Σ_2) is defined as

$$\begin{aligned} H(\Sigma_2|Q_1) &\triangleq \sum_{q_1 \in Q_1} P(q_1)H(\Sigma_2|q_1) \\ &= - \sum_{q_1 \in Q_1} \sum_{\sigma_1^2 \in \Sigma_2} P(q_1)P(\sigma_1^2|q_1) \log P(\sigma_1^2|q_1) \end{aligned} \quad (4)$$

where $P(q_1)$ is the probability of a PFSA state $q_1 \in Q_1$ and $P(\sigma_1^2|q_1)$ is the conditional probability of a symbol $\sigma_1^2 \in \Sigma_2$ given that a PFSA state $q_1 \in Q_1$ is observed. Figure 1 shows a cross transition at time t from $\{s_1\}$ to $\{s_2\}$ for varying depths of \mathcal{M}_1 .

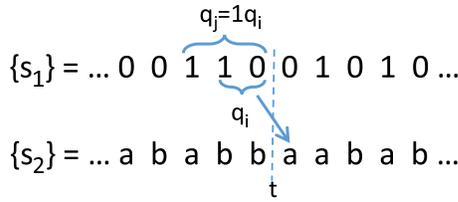


Fig. 1. Variable depth xD -Markov machine

In xD -Markov machines, a symbol block of (finite) length D is sufficient to describe the current state for the PFSA constructed from the symbol stream $\{s_1\}$. In other words, the symbols that occur prior to the last D symbols do not affect the subsequent symbols observed in symbol stream $\{s_2\}$. Therefore, the number of states of a xD -Markov machine of depth D is bounded above by $|\Sigma_1|^D$, where $|\Sigma_1|$ is the cardinality of the alphabet Σ_1 . For example, with the alphabet size $|\Sigma_1| = 5$ and a depth $D = 3$, the xD -Markov machine could have at most $|\Sigma_1|^D = 125$ states. As this relationship is exponential in nature, the number of states rapidly increases as D is increased. However, from the perspective of modeling the cross-dependance from $\{s_1\}$ to $\{s_2\}$, some states may be more important than others in terms of their embedded causal information contents. Therefore, it is advantageous to have a set of states that correspond to symbol blocks of different lengths. This is accomplished by starting off with the simplest set of states (i.e., $Q_1 = \Sigma_1$ for $D = 1$) and subsequently splitting the current state that results in the largest decrease of the cross entropy rate $H(\Sigma_2|Q_1)$ (see Eq. (4)). This underlying procedure is called *state splitting* [12]. This way of creating the state space restricts the exponential growth of states with increasing depth D .

The process of splitting a state $q_1 \in Q_1$ is executed by replacing the symbol block q_1 by its *branches* as described by the set $\{\sigma_1^1 q_1 : \sigma_1^1 \in \Sigma_1\}$ of words, where $\sigma_1^1 q_1$ represents the equivalence class of all (finite-length) symbol strings with the word q_1 as the suffix. Figure 2 illustrates the process of state splitting in a PFSA with alphabet $\Sigma_1 = \{0, 1\}$, where each terminal state is circumscribed by

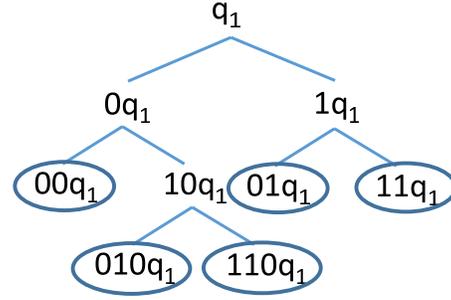


Fig. 2. Tree-representation of state splitting in the PFSA \mathcal{M}_1

an ellipse. For example, the states in the third layer from the top are: $00q_1$, $10q_1$, $01q_1$, and $11q_1$, of which all but $10q_1$ are terminal states. Consequently, the state $10q_1$ is further split as $010q_1$ and $110q_1$ that are also terminal states, i.e., $Q = \{00q_1, 01q_1, 11q_1, 010q_1, 110q_1\}$, as seen in the split PFSA diagram of Figure 2. Maximum reduction of the cross entropy rate is the governing criterion for selecting the state to split. In addition, the generated set of states must satisfy the self-consistency criterion for the PFSA \mathcal{M}_1 , which only permits a unique transition to emanate from a state for a given symbol. If $\delta(q_1, \sigma_1^1)$ is not unique for each $\sigma_1^1 \in \Sigma_1$, then the state q is split further. In the state splitting algorithm, a stopping rule is constructed by specifying the threshold parameter η_{spl} on the rate of decrease of cross entropy rate. An alternative stopping rule for the algorithm is to provide a maximal number of states N_{max} of PFSA \mathcal{M}_1 instead of the threshold parameter η_{spl} .

At each step of state splitting, each element $\pi_{12}(q_1, \sigma_1^2)$ of the cross morph matrix Π_{12} is estimated by frequency counting as the ratio of the number of times, $N(q_1 \sigma_1^2)$, the state q_1 from $\{s_1\}$ is followed by the symbol σ_2 from $\{s_2\}$ and the number of times, $N(q_1)$, the state q_1 occurs. Each element $\hat{\pi}_{12}(q_1, \sigma_1^2)$ of the estimated morph matrix $\hat{\Pi}_{12}$ is obtained [12] as

$$\hat{\pi}_{12}(q_1, \sigma_1^2) \triangleq \frac{1 + N(q_1 \sigma_1^2)}{|\Sigma_2| + N(q_1)} \quad \forall \sigma_1^2 \in \Sigma_2 \quad \forall q_1 \in Q_1 \quad (5)$$

where $\sum_{\sigma_1^2 \in \Sigma_2} \hat{\pi}_{12}(q_1, \sigma_1^2) = 1 \quad \forall q_1 \in Q_1$.

Similar to Eq. (5) each element $P(q_1)$ of the stationary state probability vector for the PFSA from $\{s_1\}$ at a certain splitting stage is estimated by frequency counting [12] as

$$\hat{P}(q_1) \triangleq \frac{1 + N(q_1)}{|Q_1| + \sum_{q_1' \in Q_1} N(q_1')} \quad \forall q_1 \in Q_1 \quad (6)$$

where $\hat{P}(q_1)$ is an element of the estimated stationary state probability vector, which implies the estimated stationary

probability of the PFSA being in the state $q_1 \in Q_1$. Now the cross entropy rate (see Eq. (4)) is computed in terms of the elements of estimated state probability vector for the PFSA from first time series and estimated cross morph matrix as

$$\begin{aligned} H(\Sigma_2|Q_1) &= - \sum_{q_1 \in Q_1} \sum_{\sigma_1^2 \in \Sigma_2} P(q_1)P(\sigma_1^2|q_1) \log P(\sigma_1^2|q_1) \\ &\approx - \sum_{q_1 \in Q_1} \sum_{\sigma_1^2 \in \Sigma_2} \hat{P}(q_1)\hat{\pi}_{12}(q_1, \sigma_1^2) \log \hat{\pi}_{12}(q_1, \sigma_1^2) \end{aligned} \quad (7)$$

Based on the specific threshold, the process of splitting is continued till the optimal cross model is achieved. The final estimated morph matrix $\hat{\Pi}_{12}$ is used as a representative feature of the causality from first to second time-series.

4. DESCRIPTION OF DATA ACQUISITION

A brand new (12V AGM VRLA with 56Ah capacity) lead-acid battery has been used in the experiments [18]. As the battery is charged/discharged according to given input (current) profiles at the room temperature and an ensemble of synchronized time-series of the input charge/discharge current and output voltage responses is collected at the sampling frequency of 1Hz. A typical input current profile for this experiment is depicted in Fig. 3. The duration of a input profile is ~ 150 hours, which consists of three capacity measure cycles that are followed by 25 duty cycles.

A capacity measurement cycle is a slow, full discharge/discharge cycle, where the battery is fully discharged followed by a full charge. In this constant-current constant-voltage (CCCV) operation of the experiments, the battery is discharged by a constant current of $-20A$ for ~ 2 hours at first. Then it is charged first by constant current of $20A$ until its output reaches a voltage of $13.8V$; this voltage is kept constant for the next 3 hours with gradually decreasing charging current. The maximum capacity at that time is measured by integrating the current during the charge period. Three computed battery maximum capacities are obtained from these capacity measurement cycles, the mean value of them is considered as the nominal maximum capacity for that particular time. Since there are five similar (i.e., same pattern) input profiles in total are applied to the battery during the whole experiment. The degradation of battery SOH (i.e., the ratio of maximum capacity between "now" and when it is brand new) is obtained.

Total 25 duty cycles are divided into groups of five. The transition time between two consecutive groups is ~ 6 hours with charging to full capacity, while the transition time between two consecutive duty cycles in one group is

~ 1.2 hours with inadequate recharging. Each duty cycle last ~ 2.7 hours, which is composed of ~ 75 "Hotel-Pulse" cycles, as depicted in Fig. 3(b). Each individual "Hotel-Pulses" cycle (i.e., duration of 120s) consists of a "hotel" load (i.e., relatively steady discharge due to "hotel" needs like lighting and other electrical equipments) and a discharge pulse followed by a charge (i.e., regeneration) pulse, as shown in Fig. 3(a). The amplitude of the "hotel" load and the discharging & charging pulses are not usually monotonically increasing in a duty cycle, which makes each duty cycle slightly different from others. This pattern of input cycles largely simulates a real-time working condition for an electric locomotive. Further details of the time-series data characteristics could be found in [18].

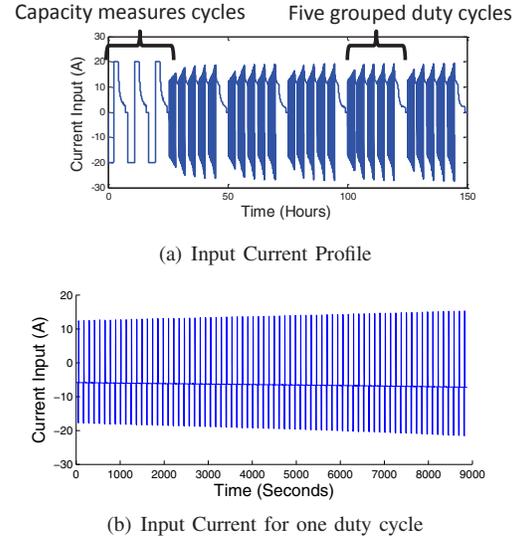


Fig. 3. Profile of input current data for the experiment

5. xD-MARKOV MODELING

Time-series data for both the input and output are first normalized individually by subtracting the mean and dividing by the standard deviation of their elements; this step corresponds to bias removal and variance normalization. The input and output data are normalized in a moving window fashion to get rid of any trend or drooling behavior in the data. Then, a wavelet-based segmentation [18] is done to extract relevant segments of the data based on their frequency content.

Data from engineering systems is typically oversampled to ensure that the underlying dynamics can be captured. Due to coarse-graining from the symbolization process, an over-sampled time-series may mask the true nature of the system dynamics in the symbolic domain (e.g., occurrence of self loops and irrelevant spurious transitions in the xD-Markov machine). Time-series is first down-sampled to

find the next crucial observation. The first minimum of the absolute auto-correlation function generated from the observed time-series is obtained to find the uncorrelated samples in time. The data sets are then down-sampled by this lag. The time lag for the output data is used to downsample both the input and output data. The rationale is to keep the data synchronous and not miss the relevant dynamics of the observed output. To avoid discarding significant amount of data due to downsampling, down-sampled data using different initial conditions is concatenated. Further details of this pre-processing can be found in [19], [20].

The down-sampled time-series data set is then partitioned using maximum entropy partitioning (MEP) [16]. A ternary alphabet A (i.e., $|A| = 3$) has been used to symbolize both the input and output data individually. Clearly, the choice of the partitioning method for symbolization will affect the ability of the xD -Markov machine $\mathcal{M}_{1 \rightarrow 2}$ to resolve the dynamics of the underlying cross-dependence. The choice of the partitioning method is dependent on the goal of the modeling process, readers are referred to survey papers [21], [22] for more details on the choice of symbolization. Work presented here deals only with the task of modeling the cross-dependence between the symbol sequences given that some technique has already been used for discretization of the observed time-series. Once the data are symbolized, the state splitting algorithm presented earlier in section 3-B is applied to infer the variable depth (D) of the cross model from input current to output voltage for different values of SOH.

6. RESULTS AND INTERPRETATION

This section presents the results to demonstrate the efficacy of the proposed method for health monitoring of a lead-acid battery.

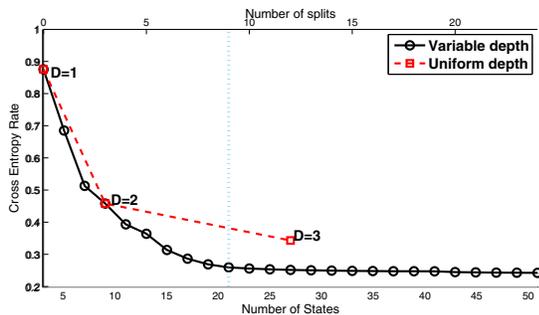


Fig. 4. Cross entropy rate vs number of splits (SOH = 1, 3 symbols, downsample time-lag = 7).

Figure 4 shows the behavior of the change in cross entropy rate (see Eq. (4)), that is defined from input current to output voltage, with an increasing number of state splitting in the PFSA constructed from input current with alphabet A . The output voltage symbol sequence has

also the same cardinality. The analysis is carried out at the perfect health of the battery i.e., SOH = 1. Figure 4 shows a monotonic decrease in the cross-entropy rate with negligible improvement after the splitting tree contains a certain number of states (implying convergence). In the absence of a competing objective to minimize the complexity of the model obtained by splitting, the state-splitting is terminated when addition of new states does not significantly improve the cross-entropy rate. After the 9th split at the state space of cardinality 21 (denoted by vertical dotted line), the entropy rate improvement becomes negligible as shown in Figure 4. This particular state space is chosen subsequently for modeling the cross dependence at different levels of SOH. Figure 4 also shows that the cross entropy rate for uniform depth scenario decreases with a lower rate than the variable depth scenario proposed here.

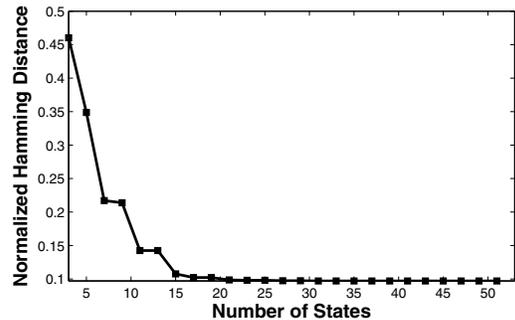


Fig. 5. Normalized Hamming distance vs number of states (SOH = 1, 3 symbols, downsample time-lag = 7)

The accuracy of the model obtained by state-splitting could be analyzed by using it for making predictions on the output (i.e., voltage) symbol sequence based on the observations on the input (i.e., current) states. The xD -Markov model is first learned from 50% of the time-series of current and voltage with the state splitting algorithm to construct crossed PFSA with different states (obtained as trees of different lengths). All the models are then used to predict the remaining output symbol sequence (i.e., from the remaining 50% voltage time-series) using the remaining input symbol sequence (i.e., from the remaining 50% current time-series). Upon observation of an input state, a maximum likely prediction on the output symbol is made using the estimated morph matrix. The error in prediction is then measured as the normalized Hamming distance between the actual and the predicted test results for the voltage symbol sequence which is shown in Figure 5. The prediction error monotonically decreases and finally saturates (i.e., no further reduction with extra states) after the model with 21 states (after 9 splits) is reached, which was earlier inferred using cross entropy rate.

A change in battery health (i.e., SOH) should cause a

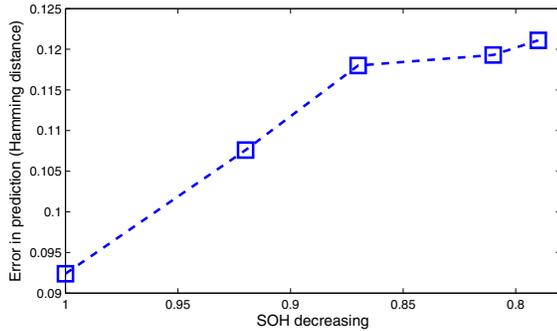


Fig. 6. Error in prediction of output with decreasing health)

departure in the input-output behavior from the maximum-likely behavior of the input-output model inferred at perfect health, i.e., for $SOH = 1$. To see this, the model learned at $SOH = 1$ is used to predict the maximum-likely voltage symbol sequence based on observations on current state-space for different health conditions. Figure 6 shows variations in the prediction error with deteriorating health of the battery measured as normalized Hamming distance. The prediction error, based on the state space from the xD -Markov model at $SOH = 1$, increases monotonically as the battery health deteriorates (i.e., SOH drops). Hence, this prediction error can be used as a possible indicator of SOH degradation.

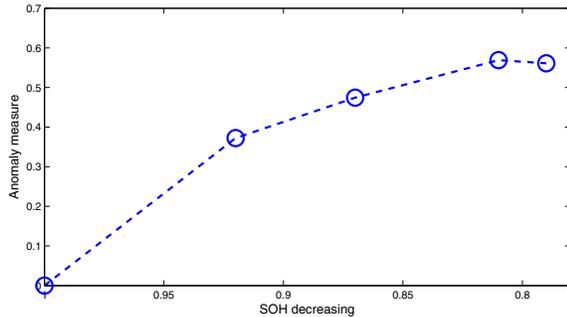


Fig. 7. Anomaly measure vs state of health (SOH)

An anomaly measure for reflecting SOH degradation is formulated based on the xD -Markov model at the corresponding health condition of the battery. The xD -markov machine at $SOH = 1$ is considered the reference condition. The measure at an SOH level is defined as the difference in Frobenius norms of the estimated morph matrix at that SOH level and that of the estimated morph matrix at $SOH = 1$. The morph matrices are estimated by the approach that is shown in section 3. Based on the analysis, presented in Figure 4, the stopping criteria for state splitting is chosen to be 21. Figure 7 presents the

variation of the proposed anomaly measure with respect to decreasing SOH . The proposed measure (nearly) monotonically increases with decreasing SOH . It is noted that the model used for estimating anomaly was inferred without tying the objective function with any performance measure (like anomaly detection or class separability). It is possible to supervise the model inference depending on the tasks like anomaly detection, classification etc. and get a better behavior than that shown in Figure 7.

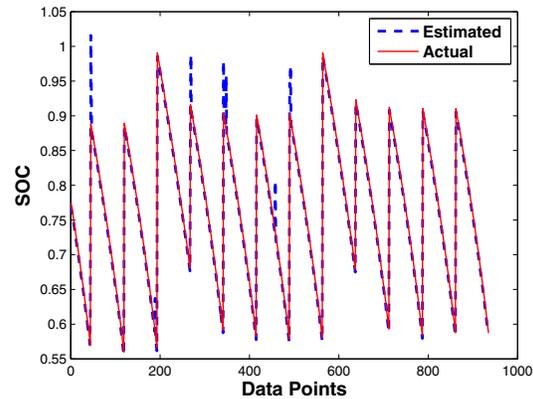


Fig. 8. SOC estimation using the regression-based filter. $SOH=1$

Finally, a regression-based filter is designed to estimate the SOC for the battery. During a training phase, the expected change in the SOC is learned using a k-Nearest Neighbor-based (kNN) [23] regression using the input-output data. For feature extraction, the input-output morph matrices are first estimated using time-series data for 20 minutes in a sliding window fashion. For each calculation, a new "Hotel-Pulse" (see section 4. Pulse duration is 2 minutes) is added to the window while the last pulse is pushed out. It is noted that the predictions are made every two minutes (which is the original frequency of SOC measurement during experiments). 50% of the data is used for training while the remaining half is used for test. For computational efficiency and limited data (the probabilities need significant amount of data for convergence), the state-splitting for xD -Markov machine construction is terminated after 7 states with $|A| = 3$ (i.e., the morph matrix having 21 elements). To further reduce the dimensionality of the feature space, the top 8 (out of the 21, $\sim 40\%$) features are selected by using Principal Component Analysis (PCA) [23]. These features are then used to learn the kNN-based regression with $k = 4$. During test, the regression is used to make an estimate on the expected change in the battery SOC. With the previous SOC known, this is used to predict the current SOC for the battery. The estimation results are presented in Figure 8 that shows a near-perfect estimation of SOC using the proposed filter. The average absolute error in the prediction

of SOC is $\sim 0.6\%$.

7. SUMMARY AND CONCLUSIONS

A dynamic data-driven symbolic analysis technique is proposed in this paper to capture the causal cross-dependence directed from input to output in dynamical systems. The generative model for input-output dependence is called xD -Markov machine, which is constructed based on the algebraic structure of probabilistic finite state automata (PFSA). The state space of the input PFSA in the xD -Markov machine is obtained via state splitting after independent symbolization of both input and output time-series. The proposed approach essentially captures the cross-dependence towards output without causing an exponential growth in cardinality of the state space. The method is validated on a set of charging-discharging data of lead-acid batteries at different SOH conditions. It is shown that the cross entropy rate, which is the objective function for state splitting, drops drastically in the first few levels of state splitting and subsequently gradually converges. Different types of stopping criteria for state splitting are described based on the validation data. The predictability of the output data from observing the input data is also tested for different levels of splitting in the xD -Markov model construction. The variation of the prediction error with respect to a growing extent of state-splitting shows a similar nature as the cross entropy rate. The prediction error based on the xD -Markov model at $SOH = 1$, which is formulated as the Hamming distance between actual and predicted output symbol sequences, increases monotonically as the health of the battery deteriorates (i.e., as SOH decreases from 1). An anomaly measure is also constructed based on the morph matrix of corresponding xD -Markov machine to predict SOH online. A regression-based filter is designed on the morph matrices of xD -Markov machines, which yields in a SOC estimation error of only $\sim 0.6\%$.

ACKNOWLEDGEMENTS

The authors are thankful to Prof. Christopher D. Rahn for technical consultation as well as for providing the battery data.

REFERENCES

- [1] C. W. J. Granger, "Economic processes involving feedback," *Information and control*, vol. 6, no. 1, pp. 28–48, 1963.
- [2] D. Heckerman, "A bayesian approach to learning causal networks," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 285–295, Morgan Kaufmann Publishers Inc., 1995.
- [3] C. J. Quinn, T. P. Coleman, N. Kiyavash, and N. G. Hatsopoulos, "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings," *Journal of computational neuroscience*, vol. 30, no. 1, pp. 17–44, 2011.
- [4] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhat-tacharya, "Causality detection based on information-theoretic approaches in time series analysis," *Physics Reports*, vol. 441, no. 1, pp. 1–46, 2007.
- [5] C. W. Granger, "Testing for causality: a personal viewpoint," *Journal of Economic Dynamics and control*, vol. 2, pp. 329–352, 1980.
- [6] P. E. Caines and C. Chan, "Feedback between stationary stochastic processes," *Automatic Control, IEEE Transactions on*, vol. 20, no. 4, pp. 498–508, 1975.
- [7] M. Gevers and B. Anderson, "On jointly stationary feedback-free stochastic processes," *Automatic Control, IEEE Transactions on*, vol. 27, no. 2, pp. 431–436, 1982.
- [8] V. Solo, "On causality and mutual information," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 4939–4944, IEEE, 2008.
- [9] J. Rissanen and M. Wax, "Measures of mutual and causal dependence between two time series (corresp.)," *Information Theory, IEEE Transactions on*, vol. 33, no. 4, pp. 598–601, 1987.
- [10] C. Gourieroux, A. Monfort, and E. Renault, "Kullback causality measures," *Annales d'Economie et de Statistique*, pp. 369–410, 1987.
- [11] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.
- [12] 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.
- [13] A. Srivastav, Y. Wen, E. Hendrick, I. Chattopadhyay, A. Ray, and S. Phoha, "Information fusion for object & situation assessment in sensor networks," in *American Control Conference (ACC), San Francisco, CA, USA*, pp. 1274–1279, June-July 2011.
- [14] L. K. Saul and M. I. Jordan, "Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones," *Machine learning*, vol. 37, no. 1, pp. 75–87, 1999.
- [15] S. Sarkar, S. Sarkar, N. Virani, A. Ray, and M. Yasar, "Sensor fusion for fault detection and classification in distributed physical processes," *Frontiers in Robotics and AI*, vol. 1, p. 16, 2014.
- [16] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Processing*, vol. 86, pp. 3309–3320, November 2006.
- [17] S. Sarkar, K. Mukherjee, X. Jin, D. S. Singh, and A. Ray, "Optimization of symbolic feature extraction for pattern classification," *Signal processing*, vol. 92, no. 3, pp. 625–635, 2012.
- [18] Y. Li, P. Chattopadhyay, A. Ray, and C. Rahn, "Identification of the battery state-of-health parameter from input-output pairs of time series data," *Journal of Power Sources*, doi: 10.1016/j.jpowsour.2015.03.068.
- [19] A. Srivastav, "Estimating the size of temporal memory for symbolic analysis of time-series data," in *American Control Conference (ACC), 2014*, pp. 1126–1131, June 2014.
- [20] D. Jha, A. Srivastav, K. Mukherjee, and A. Ray, "Depth estimation in markov models of time-series data via spectral analysis," in *Preprints American Control Conference (ACC), Chicago, IL, USA*, 1-3 July 2015.
- [21] H. Liu, F. Hussain, C. Tan, and M. Dash, "Discretization: An enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, pp. 393–423, 2002.
- [22] S. Garcia, J. Luengo, J. A. Saez, V. Lopez, and F. Herrera, "A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. PrePrints, 2012.
- [23] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006.