2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

For presentation in American Control Conference 2013, Washington DC

Maximally Bijective Discretization for Data-driven Modeling of Complex Systems

Soumik Sarkar sarkars@utrc.utc.com
Abhishek Srivastav srivasa1@utrc.utc.com
Madhusudana Shashanka shasham@utrc.utc.com
Decision Support & Machine Intelligence Group, Systems Department

United Technologies Research Center
East Hartford, CT 06108, USA

Keywords: *Time series Discretization, Dynamical Systems, Symbolic Modeling*

Abstract

Phase-space discretization is a necessary step for study of continuous dynamical systems using a language-theoretic approach. It is also critical for many machine learning techniques, e.g., probabilistic graphical models (Bayesian Networks, Markov models). This paper proposes a novel discretization method – Maximally Bijective Discretization, that finds a discretization on the dependent variables given a discretization on the independent variables such that the correspondence between input and output variables in the continuous domain is preserved in discrete domain for the given dynamical system.

1. Introduction

Discretization - the process of coarse-graining of the data-space under some notion of similarity, transforms continuous valued variables to a discrete symbolic space. This is an important pre-processing step for several applications such as *symbolic time-series analysis* of dynamical systems, data-mining and machine learning. Given a continuous data-space, a compact region of interest is partitioned into a finite number of mutually exclusive and exhaustive *cells* or *partitions* and each partition is associated with a symbol. The time evolution of a dynamical system can now be studied in the symbolic space using a language-theoretic approach with tools such as shift-maps and sliding block codes in both deterministic and probabilistic setting. On the other hand, for many applications in data-mining and machine learning, discretization is necessitated by the choice of tools that work only or are computationally more efficient with discrete data.

Despite the necessity and importance of discretization, there is no standard way to approach it. This is because several factors such as the nature of the dynamical system or data set in question, choice of the similarity metric, and desired model simplicity affect the nature of a discretization scheme that is appropriate. Moreover, even if one can define the optimality criteria for discretization, the process will be NP-complete [1]. As pointed out by [2], while discretization is desirable pre-processing step, practical discretization schemes are necessarily heuristic in nature and a large number of such schemes have been proposed in the literature. The simplest methods include *equal interval width* (uniform) and *equal interval frequency* (maximum entropy) discretization or partitioning. Symbolic false nearest neighbor partitioning (SFNNP) [3] optimizes a generating partition by avoiding topological degeneracy. However, SFNNP may become extremely computation and memory intensive if the dimension of the phase space of the underlying dynamical system is large and the noise content of the data is high. The wavelet transform largely alleviates the above shortcoming and is particularly effective with noisy data for large-dimensional dynamical systems [4]. Subbu, Ray [5] and Sarkar et. al [6] introduced a Hilbert-transform-based analytic signal space partitioning (ASSP) as an alternative to the wavelet space partitioning (WSP).

This paper proposes a supervised discretization

978-1-4799-0178-4/$31.00 ©2013 AACC

scheme for symbolic modeling of dynamical systems [7]. The goal is to find a discretization of the output space given some partition on the input space of a dynamical system such that the input-output dynamics of the system is preserved in the symbolic domain. For this purpose, we propose a Maximally Bijective Discretization scheme that aims to maximize the input-output symbol correspondence for a given dynamical system supervised by *reward* function.

2. Background and Motivation

Some complex system behaviors of interest can be faithfully represented by a dynamical system in the continuous time and in continuous state space. However, it is often difficult to obtain explicit functional relationships from data in the continuous domain for many real complex systems. Discrete symbolic modeling is preferred in such situations where the primary goal is to preserve the functional relationships of the underlying continuous domain system.

2.1. Representation of Dynamical Systems

In the context of symbolic system identification, the underlying structure of a dynamical system is represented by a Generalized Dynamical System (*GDS*) [8]. A simplified definition is as follows:

Definition 2.1 *A Generalized Dynamical System (*GDS*) can be defined as an* 6*-tuple automaton.*

$$D = (T, U, W, Q, f, g) \tag{1}$$

where

- T *is a time set (e.g.* $T = [0, \infty)$*),*
- U *and* W *are input and output sets respectively,*
- Q *are internal states,*
- f *denotes the global state transition*

$$f : T \times Q \times U \to Q \quad \textit{for time-varying systems} \tag{2}$$

$$f : Q \times U \to Q \quad \textit{for time-invariant systems} \tag{3}$$

- g *denotes the output function*

$$g : T \times Q \to W \quad \textit{for time-varying systems} \tag{4}$$

$$g : Q \to W \quad \textit{for time-invariant systems} \tag{5}$$

For data-driven symbolic system identification of a complex system, input and output variables are discretized temporally and spatially to generate blocks of symbols, also called *words*. A grammar is the mathematical structure that constrains the inter-relationship among these words. Let the quantized abstraction of the *GDS* is called a Qualitative Dynamical System (*QDS*).

Definition 2.2 *A Qualitative Dynamical System (*QDS*) can be represented as a* 5*-tuple*

$$\mathscr{G} = \{\mathscr{Q}, \Lambda, \Sigma, \delta, \gamma\} \tag{6}$$

where

- $\mathscr{Q} \triangleq \{q_1, q_2, ..., q_f\}$ *is the finite set of qualitative states of the automaton.*
- $\Lambda \triangleq \{\lambda_1, \lambda_2, ..., \lambda_m\}$ *is the set of qualitative input events.*
- $\Sigma \triangleq \{\sigma_1, \sigma_2, ..., \sigma_n\}$ *is the set of output alphabets.*
- $\delta : \mathscr{Q} \times \Lambda \to \mathscr{Q}$ *is the state transition function that maps the current state into the next state upon receiving the input* λ*. The state transition function can be stochastic; in that case,*

$$\delta : \mathscr{Q} \times \Lambda \to Pr\{\mathscr{Q}\} \tag{7}$$

where, $Pr\{\mathscr{Q}\}$ *is a probability distribution over* $\mathscr{Q}$*.*

- $\gamma : \mathscr{Q} \to \Sigma$ *is the output generation function that determines the output symbol from the current state.* γ *can be stochastic as well, i.e., (with similar notation as before)*

$$\gamma : \mathscr{Q} \to Pr\{\Sigma\} \tag{8}$$

2.2. Abstraction of *GDS*

Abstraction is the process of transforming a general dynamical system into its qualitative counterpart. The method is formalized as follows: Let χ denote a set of qualitative abstraction functions

$$\chi : D \to \mathscr{G} \tag{9}$$

χ consists of three individual abstraction functions:

$$\chi = (\chi_{TQU}, \chi_Q, \chi_W), \quad \text{where}$$

$$\chi_{TQU} : T \times Q \times U \to \Lambda \tag{10}$$

$$\chi_Q : Q \to \mathscr{Q} \tag{11}$$

$$\chi_W : W \to \Sigma \tag{12}$$

Kokar [8] introduced a set of necessary and sufficient conditions, or 'consistency postulates' that the pair $\mathscr{G},\chi$ must satisfy in order to be a valid representation of the general dynamical system. The consistency postulates can be stated as follows:

Definition 2.3 *Let D, $\mathscr{G}$ and χ represent a GDS, QDS and an abstraction function respectively. Then the pair $(\mathscr{G},\chi)$ forms a consistent representation in a probabilistic sense if, $\forall q,u,t$,*

$$\gamma(\chi_Q(q)) = \chi_W(g(q)) \tag{13}$$

$$\chi_Q(f(t,q,u)) \sim \delta(\chi_Q(q), \chi_{TQU}(t,q,u)) \tag{14}$$

where $X \sim P$ means the random variable X is distributed according to the probability distribution P.

Theorem 2.1 (Kokar [8]) *Let $W_\pi = W_1,...,W_n,\ n \in \mathbb{N}$ be a finite discretization of a GDS's output space W, given by $\chi_W^{-1} : \Sigma \to W_\pi$. Let Q_π describe a discretization of Q defined as an inverse image of W_π through g,*

$$Q_\pi = g^{-1}(W_\pi),$$

and let TQU_π describe a discretization of $T \times Q \times U$ defined as an inverse image of Q_π through f,

$$TQU_\pi = f^{-1}Q_\pi.$$

Then Q_π is a maximal admissible discretization of Q, and TQU_π is an admissible discretization of $T \times Q \times U$.

With such discretization, the *QDS* is related to the *GDS* through a homomorphism. If the system model of the *GDS* is known, such discretization can be analytically evaluated and utilized as delineated above. However, in the absence of model equations, alternate ways are required to evaluate such discretization without explicitly knowing functions f and g.

One possible scheme [7] involves construction of a pseudo phase space from the output signal using Taken's theorem [9]. After such construction, the pseudo phase space and the input space are individually discretized. The main idea of this scheme is to place the boundaries of the discretization segments in such a way, that a change in both input and output symbols is synchronized. A discretization constructed in this way is admissible, but may not be maximal, since this discretization is a sub-discretization of the original discretization proposed in Theorem 2.1. The discretization scheme is illustrated in Fig. 1 and Fig. 2. In this example, a simulated system is considered with input/output

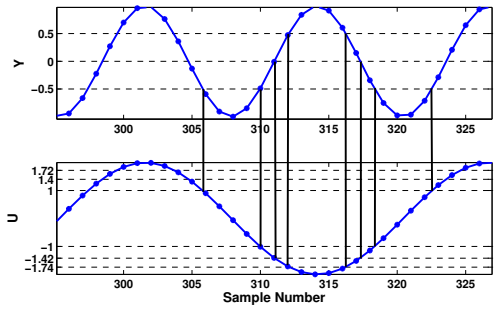

Figure 1. Admissible discretization scheme showed on an illustrative example system; Bin boundaries are marked on corresponding axes as grid lines

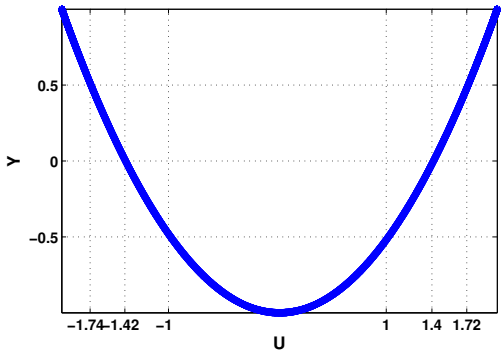

Figure 2. Scatter plot of input U and output Y with admissible discretization of U given uniform discretization of Y; Bin boundaries are marked on corresponding axes as grid lines

signals as:

$$U(t) = 2\cos(0.25t) \tag{15}$$

$$Y(t) = \cos(0.5t) \tag{16}$$

It is clear that periodicity (or at least quasi-periodicity) guarantees that the number of output and input symbols will not explode. Also, evaluation of such discretization may not be so obvious and the alphabet size may explode when the data is significantly noisy. This is the prime motivation of this present work where the following crucial observation is made: the above discretization process essentially aims to maximize the degree of input-output symbol correspondence.

3. Methodology and Algorithm

This section develops the methodology and algorithm for a discretization scheme that maximizes the

degree of input-output symbol correspondence, hence named Maximally Bijective Discretization (MBD).

3.1. Representation of Discretization

Let the time series data of a variable generated from a physical complex system or its dynamical model be denoted as $\mathbf{q}$. A compact (i.e., closed and bounded) region $\Omega \in \mathbb{R}^n$, where $n \in \mathbb{N}$, within which the time series is circumscribed, is identified. Let the space of time series data sets be represented as $\mathbf{Q} \subseteq \mathbb{R}^{n \times N}$, where the $N \in \mathbb{N}$ is sufficiently large for convergence of statistical properties within a specified threshold. Note, n represents the dimensionality of the time-series and N is the number of data points in the time series. A discretization encodes Ω by introducing a partition $\mathbb{B} \equiv \{B_0, ..., B_{(p-1)}\}$ consisting of p mutually exclusive (i.e., $B_j \cap B_k = \emptyset \ \forall j \neq k$), and exhaustive (i.e., $\cup_{j=0}^{p-1} B_j = \Omega$) cells. For one-dimensional time series data, a discretization consisting of p cells is represented by $p-1$ points that serve as cell boundaries. In the sequel, a p-cell discretization $\mathbb{B}$ is also expressed as $\Gamma_m \triangleq \{\gamma_1, \gamma_2, \cdots, \gamma_{m-1}\}$, where γ_i denotes a bin boundary.

3.2. Maximally Bijective Discretization

Let the complex system under consideration has m real-valued output variables and n real-valued input variables that are denoted as $w_1, \cdots, w_m$ and $u_1, \cdots, u_n$ respectively. In the continuous domain, an input variable u_i for any i is represented as a function of all output variables for the purpose of discretization as follows:

$$u_i = h_i(w_1, \cdots, w_m) \tag{17}$$

Suppose a m-dimensional discretization is imposed on the space of output variables that divides the data for output variables into K discrete classes. The goal is to discretize each input variable based on the defined classes of output variables. This paper develops a *Maximally Bijective* scheme of performing the discretization of each input variable separately. The problem is formulated in the sequel for any input variable u (omitting the subscript).

Let $\mathbb{B} \equiv \{B_0, ..., B_{(l-1)}\}$ be a discretization of one of the input variables, where $B_j, j \in \{0, ..., l-1\}$ is a bin of the discretization. With this setup, *correspondence* between a class and a bin is defined as follows:

$$\begin{aligned} \text{If } \ i &= \arg\max_k P(C_k | x \in B_j) \\ \text{then, } C_i &\Rightarrow B_j \end{aligned} \tag{18}$$

where, $P(\cdot)$ denotes a probability function and $x \in \mathbb{R}$. In words, class C_i corresponds to bin B_j. Note, a class C_i may correspond to more than one disjoint bins. From this perspective, a reward function is defined as follows:

$$R(\mathbb{B}|x) = \left\{ P(C_i | x \in B_j) : C_i \Rightarrow B_j \right\} \tag{19}$$

Note that this reward function signifies the notion of bijection, i.e., with higher reward, the probability of a class being corresponding to a bin increases. With this setup, the total expected reward is calculated as

$$TR(\mathbb{B}) = \int_X R(\mathbb{B}|x) P(x) dx \tag{20}$$

The goal here is to maximize this total reward function, hence the discretization scheme is called Maximally Bijective. It is clear from the formulation that maximizing the total reward function is equivalent to maximizing $R(\mathbb{B}|x)$ at each x.

$$\begin{aligned} \mathbb{B}^* &= \arg\max_{\mathbb{B}} \{R(\mathbb{B}|x) \ \forall x\} \\ &= \arg\max_{\mathbb{B}} \left\{ P(C_i | x \in B_j) : C_i \Rightarrow B_j \ \forall x \right\} \end{aligned} \tag{21}$$

This observation leads to an algorithm that is developed and used to identify MBD in this paper. Let $P_m(x)$ denotes $\left\{ P(C_i | x \in B_j) : C_i \Rightarrow B_j \right\}$ at any x. Note, by this definition

$$P_m(x) = \max_i P(C_i|x) \ \forall x \tag{22}$$

With this setup an overview of the proposed algorithm is provided below:

Overview of the Algorithm

$x = \min(D)$ (D denotes one dimensional data vector)
$k = 1$
while $x < \max(D)$ **do**
  Identify C_i such that $P(C_i|x) = P_m(x)$;
  Identify C_j such that $P(C_j|x+dx) = P_m(x+dx)$;
  if $i \neq j$ **then**
    $\gamma_k = x$ % Note, γ_k denotes the k^{th} bin boundary
    $k \leftarrow k+1$
  end if
  $x \leftarrow x + dx$
end while

However, the primary issue with the above process is reliable estimation of $P(C_i|x)$ from data [2]. This paper adopts a basic frequency counting (over an interval) method to estimate this conditional probability. This means that the optimization process begins with considering a small window in the domain (around $\min(D)$) to identify the most probable class in that interval. Then

the window is slided across the data range of the input variable and bin boundaries (γ_ks) are placed where the most probable class changes from one interval to the next. However, this approximation may result in sub-optimality of the solution. The rationale is as follows: Let $[a,b] \subset \mathbb{R}$ be an interval over which the frequencies of different classes are estimated to identify the most probable class. The frequency of class C_i in the interval $[a,b]$ is denoted as $n(C_i)|_{[a,b]}$ and it can be represented as

$$n(C_i)|_{[a,b]} = \int_a^b P(C_i|x)P(x)dx \tag{23}$$

To identify the most probable class in $[a,b]$, one needs to compare $n(C_i)|_{[a,b]}$ with $n(C_j)|_{[a,b]}$. However, it is known that

$$\begin{aligned} \int_a^b P(C_i|x)P(x)dx &\gtrless \int_a^b P(C_j|x)P(x)dx \\ \nRightarrow P(C_i|x) &\gtrless P(C_j|x)\ \forall x \in [a,b] \end{aligned} \tag{24}$$

Consequently, the solution may become suboptimal due to the consideration of intervals of finite width. Also, in a realistic setting, the intervals need to be wide enough to avoid significant effects of noise in the data. However, the following lemma can be stated in this scenario:

Lemma 3.1 *The total reward (TR) is a nondecreasing function of adding new bin boundaries.*

This ensures that the total reward at least does not reduce when a new bin boundary is introduced in the sequential algorithm proposed here. A rough proof sketch of this property is provided here:

Proof Sketch: First of all, it should be noted that introduction of a new bin boundary can be considered as splitting one of the current bins. Let B_j be the bin which is splitted into B_j^1 and B_j^2 with the introduction of a new bin boundary. Also, let class C_i be the most probable class of the bin B_j with frequency $n(C_i)|_{B_j}$. Let the total reward function before and after splitting be denoted as $TR(k)$ and $TR(k+1)$ respectively. Now, three cases are possible with this splitting.

- *Case I:* In both new bins B_j^1 and B_j^2, C_i is no longer the most probable class and the most probable classes in B_j^1 and B_j^2 are say C_p and C_q respectively. In this case,

$$\begin{aligned} TR(k+1) = n(C_p)|_{B_j^1} + n(C_q)|_{B_j^2} > \\ n(C_i)|_{B_j^1} + n(C_i)|_{B_j^2} = TR(k) \end{aligned} \tag{25}$$

- *Case II:* In one of the new bins, say in B_j^1, C_i is still the most probable class. However, in B_j^2, C_q is the most probable class. In this case,

$$\begin{aligned} TR(k+1) = n(C_i)|_{B_j^1} + n(C_q)|_{B_j^2} > \\ n(C_i)|_{B_j^1} + n(C_i)|_{B_j^2} = TR(k) \end{aligned} \tag{26}$$

- *Case III:* In both new bins, C_i is still the most probable class. In this case,

$$TR(k+1) = n(C_i)|_{B_j^1} + n(C_i)|_{B_j^2} = TR(k) \tag{27}$$

Therefore, $TR(k+1) \geq TR(k)$, i.e., total reward does not decrease when new bin boundaries are introduced sequentially in the algorithm described above. This is also compatible with the intuitive notion of increase in reward with increase in complexity.

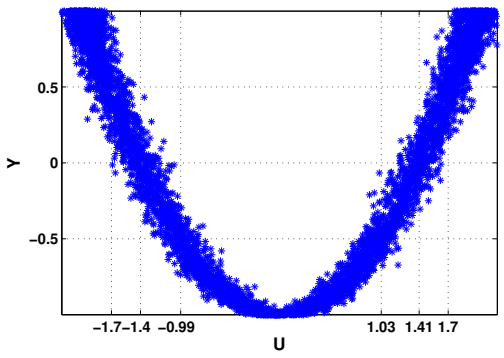

Figure 3. Illustrative example: MBD of noisy input U Given a Uniform discretization of output Y; Bin boundaries are marked on corresponding axes as grid lines

Figure 3 shows the MBD for the illustrative example problem introduced in Section 2. However, a noisy version of the data as opposed to the previous noise-free version has been used to demonstrate the efficacy of the algorithm. It is clear from the result that in a numerically stable scenario, the present approach can identify the bin boundaries of an admissible discretization defined in Section 2. Furthermore, Fig. 4 shows that the total reward monotonically increases with addition of new bin boundaries as stated in Lemma 3.1. The total reward of the MBD in this example is found to be 0.88. For comparison purposes, uniform discretization of same complexity, i.e., with same number of cells/bins (with bin boundaries $[-1.66, -1.00, -0.350.300.961.61]$) has total reward of 0.79 and maximum entropy discretization with

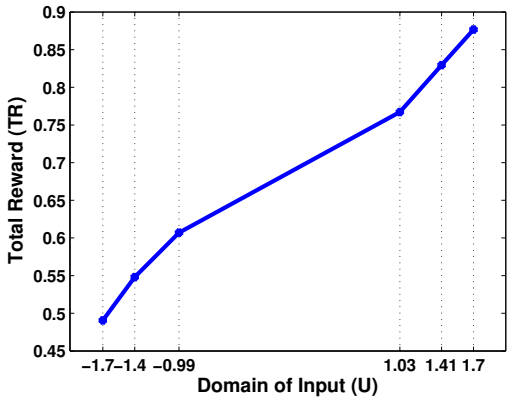

Figure 4. Monotonic increase in Total Reward with addition of new bin boundaries

same complexity (with bin boundaries $[-1.80 - 1.25 - 0.450.451.231.80]$) has total reward of 0.76. The following remark summarizes some of the key aspects the proposed discretization policy.

Remark 3.1 *The MBD scheme developed here avoids computationally expensive iterative search for bin boundaries in the data space. Instead, this algorithm scans data space once to sequentially place optimal bin boundaries and in turn identifies the optimal number of cells/bins in a stable numerical scenario. Further, the discretization of the output variables can be revised once MBD of the input variables are obtained. This process essentially involves removing certain bin boundaries of output variables that do not have effect in the MBD of the input variables.*

4. Summary, Conclusions & Future Work

This paper proposes a supervised multivariate discretization scheme for data-driven symbolic modeling of dynamical systems. The primary goal is to discretize certain dependent variables of a system given the discretization of certain independent variables such that the functional relationships among variables in the continuous domain are preserved in the discrete domain as much as possible . The primary contribution of this work is formulation and algorithm development of a MBD scheme that aims to maximize the symbolic correspondence between discretized variables. It should be noted that there are other popular cost functions (involving e.g., maximizing mutual information and minimizing Bayes risk) available in literature that may have similar effects on discretization as the current reward function proposed in this paper. Therefore, comparison of the strategy presented here with the other similar cost/reward functions is an important topic of future investigation. Other than that, the following research topics are currently being pursued as well - (1) Using discretization complexity as a competing objective to reduce the number of bins; (2) Extension of the current algorithm to handle multi-dimensional discretization; (3) Extensive validation on real data obtained from human-engineered complex systems.

References

[1] B. S. Chlebus and S. H. Nguyen, "On finding optimal discretizations for two attributes," in *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, RSCTC '98, (London, UK, UK), pp. 537–544, Springer-Verlag, 1998.

[2] Y. Yang and G. I. Webb, "Discretization for naive-bayes learning: managing discretization bias and variance," *Machine Learning*, vol. 74, pp. 39–74, 2009.

[3] M. Buhl and M. Kennel, "Statistically relaxing to generating partitions for observed time-series data," *Physical Review E*, vol. 71, no. 4, p. 046213, 2005.

[4] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, Nov 2006.

[5] A. Subbu and A. Ray, "Space partitioning via Hilbert transform for symbolic time series analysis," *Applied Physics Letters*, vol. 92, no. 8, pp. 084107–1 to 084107–3, 2008.

[6] S. Sarkar, K. Mukherjee, and A. Ray, "Generalization of Hilbert transform for symbolic analysis of noisy signals," *Signal Processing*, vol. 89, no. 6, pp. 1245–1251, 2009.

[7] S. Chakraborty, S. Sarkar, and A. Ray, "Symbolic identification for fault detection in aircraft gas turbine engines," *Proceedings of the I Mech E Part G: Journal of Aerospace Engineering*, vol. 226, no. 4, pp. 422–436, 2012.

[8] M. Kokar, "On consistent symbolic representations of general dynamic systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 8, pp. 1231–1242, 1995.

[9] F. Takens, *Detecting strange attractors in turbulence*, vol. 898/1981. Springer Berlin / Heidelberg, 1981.